Red Hat Enterprise Linux 5 Cluster Administration

Configuring and Managing a Red Hat Cluster
Edition 5

Red Hat Enterprise Linux 5 Cluster Administration

Configuring and Managing a Red Hat Cluster
Edition 5

Legal Notice

Copyright © 2014 Red Hat Inc..

This document is licensed by Red Hat under the Creative Commons Attribution-ShareAlike 3.0 Unported License. If you distribute this document, or a modified version of it, you must provide attribution to Red Hat, Inc. and provide a link to the original. If the document is modified, all Red Hat trademarks must be removed.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, MetaMatrix, Fedora, the Infinity Logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux ® is the registered trademark of Linus Torvalds in the United States and other countries.

Java ® is a registered trademark of Oracle and/or its affiliates.

XFS ® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL ® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js ® is an official trademark of Joyent. Red Hat Software Collections is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack ® Word Mark and OpenStack Logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

Abstract

Configuring and Managing a Red Hat Cluster describes the configuration and management of Red Hat cluster systems for Red Hat Enterprise Linux 5. It does not include information about Red Hat Linux Virtual Servers (LVS). Information about installing and configuring LVS is in a separate document.

Table of Contents

Introduction **4**
1. Document Conventions 4
1.1. Typographic Conventions 4
1.2. Pull-quote Conventions 6
1.3. Notes and Warnings 6
2. Feedback 7

Chapter 1. Red Hat Cluster Configuration and Management Overview **8**
1.1. Configuration Basics 8
1.1.1. Setting Up Hardware 8
1.1.2. Installing Red Hat Cluster software 9
1.1.2.1. Upgrading the Cluster Software 9
1.1.3. Configuring Red Hat Cluster Software 10
1.2. Conga 11
1.3. system-config-cluster Cluster Administration GUI 14
1.3.1. Cluster Configuration Tool 15
1.3.2. Cluster Status Tool 16
1.4. Command Line Administration Tools 17

Chapter 2. Before Configuring a Red Hat Cluster **19**
2.1. General Configuration Considerations 19
2.2. Compatible Hardware 20
2.3. Enabling IP Ports 20
2.3.1. Enabling IP Ports on Cluster Nodes 21
2.3.2. Enabling IP Ports on Computers That Run luci 21
2.4. Configuring ACPI For Use with Integrated Fence Devices 22
2.4.1. Disabling ACPI Soft-Off with chkconfig Management 23
2.4.2. Disabling ACPI Soft-Off with the BIOS 24
2.4.3. Disabling ACPI Completely in the grub.conf File 25
2.5. Considerations for Configuring HA Services 26
2.6. Configuring max_luns 29
2.7. Considerations for Using Quorum Disk 29
2.8. Red Hat Cluster Suite and SELinux 30
2.9. Multicast Addresses 31
2.10. Configuring the iptables Firewall to Allow Cluster Components 31
2.11. Considerations for Using Conga 32
2.12. Configuring Virtual Machines in a Clustered Environment 32

Chapter 3. Configuring Red Hat Cluster With Conga **34**
3.1. Configuration Tasks 34
3.2. Starting luci and ricci 34
3.3. Creating A Cluster 35
3.4. Global Cluster Properties 36
3.5. Configuring Fence Devices 39
3.5.1. Creating a Shared Fence Device 39
3.5.2. Modifying or Deleting a Fence Device 41
3.6. Configuring Cluster Members 41
3.6.1. Initially Configuring Members 41
3.6.2. Adding a Member to a Running Cluster 42
3.6.3. Deleting a Member from a Cluster 43
3.7. Configuring a Failover Domain 44
3.7.1. Adding a Failover Domain 45

3.7.2. Modifying a Failover Domain 46
3.8. Adding Cluster Resources 47
3.9. Adding a Cluster Service to the Cluster 47
3.10. Configuring Cluster Storage 50

Chapter 4. Managing Red Hat Cluster With Conga 52
4.1. Starting, Stopping, and Deleting Clusters 52
4.2. Managing Cluster Nodes 53
4.3. Managing High-Availability Services 54
4.4. Backing Up and Restoring the luci Configuration 55
4.5. Diagnosing and Correcting Problems in a Cluster 55

Chapter 5. Configuring Red Hat Cluster With system-config-cluster 56
5.1. Configuration Tasks 56
5.2. Starting the Cluster Configuration Tool 57
Custom Configure Multicast 58
Use a Quorum Disk 58
5.3. Configuring Cluster Properties 62
5.4. Configuring Fence Devices 63
5.5. Adding and Deleting Members 64
5.5.1. Adding a Member to a Cluster 64
5.5.2. Adding a Member to a Running Cluster 65
5.5.2.1. Adding a Member to a Running Cluster That Contains Only Two Nodes 65
5.5.2.2. Adding a Member to a Running Cluster That Contains More Than Two Nodes 66
5.5.3. Deleting a Member from a Cluster 67
5.5.3.1. Removing a Member from a Cluster at the Command-Line 68
5.6. Configuring a Failover Domain 69
5.6.1. Adding a Failover Domain 70
5.6.2. Removing a Failover Domain 72
5.6.3. Removing a Member from a Failover Domain 72
5.7. Adding Cluster Resources 73
5.8. Adding a Cluster Service to the Cluster 73
5.8.1. Relocating a Service in a Cluster 76
5.9. Propagating The Configuration File: New Cluster 76
5.10. Starting the Cluster Software 76

Chapter 6. Managing Red Hat Cluster With system-config-cluster 78
6.1. Starting and Stopping the Cluster Software 78
6.2. Managing High-Availability Services 78
6.3. Modifying the Cluster Configuration 80
6.4. Backing Up and Restoring the Cluster Database 81
6.5. Disabling Resources of a Clustered Service for Maintenance 82
6.6. Disabling the Cluster Software 83
6.7. Diagnosing and Correcting Problems in a Cluster 84

Example of Setting Up Apache HTTP Server 85
A.1. Apache HTTP Server Setup Overview 85
A.2. Configuring Shared Storage 85
A.3. Installing and Configuring the Apache HTTP Server 86

Fence Device Parameters 89

HA Resource Parameters 99

HA Resource Behavior 110
D.1. Parent, Child, and Sibling Relationships Among Resources 110

D.1. Parent, Child, and Sibling Relationships Among Resources 110
D.2. Sibling Start Ordering and Resource Child Ordering 111
D.2.1. Typed Child Resource Start and Stop Ordering 112
Typed Child Resource Starting Order 113
Typed Child Resource Stopping Order 113
D.2.2. Non-typed Child Resource Start and Stop Ordering 113
Non-typed Child Resource Starting Order 114
Non-typed Child Resource Stopping Order 115
D.3. Inheritance, the <resources> Block, and Reusing Resources 115
D.4. Failure Recovery and Independent Subtrees 116
D.5. Debugging and Testing Services and Resource Ordering 117

Cluster Service Resource Check and Failover Timeout 119
E.1. Modifying the Resource Status Check Interval 119
E.2. Enforcing Resource Timeouts 119
E.3. Changing Consensus Timeout 120

High Availabilty LVM (HA-LVM) 122
F.1. Configuring HA-LVM Failover with CLVM (preferred, Red Hat Enterprise Linux 5.6 and later) 123
F.2. Configuring HA-LVM Failover with Tagging 124

Upgrading A Red Hat Cluster from RHEL 4 to RHEL 5 126

Revision History 129

Index 133

Introduction

This document provides information about installing, configuring and managing Red Hat Cluster components. Red Hat Cluster components are part of Red Hat Cluster Suite and allow you to connect a group of computers (called *nodes* or *members*) to work together as a cluster. This document does not include information about installing, configuring, and managing Linux Virtual Server (LVS) software. Information about that is in a separate document.

The audience of this document should have advanced working knowledge of Red Hat Enterprise Linux and understand the concepts of clusters, storage, and server computing.

For more information about Red Hat Enterprise Linux 5, refer to the following resources:

- *Red Hat Enterprise Linux Installation Guide* — Provides information regarding installation of Red Hat Enterprise Linux 5.
- *Red Hat Enterprise Linux Deployment Guide* — Provides information regarding the deployment, configuration and administration of Red Hat Enterprise Linux 5.

For more information about Red Hat Cluster Suite for Red Hat Enterprise Linux 5, refer to the following resources:

- *Red Hat Cluster Suite Overview* — Provides a high level overview of the Red Hat Cluster Suite.
- *Logical Volume Manager Administration* — Provides a description of the Logical Volume Manager (LVM), including information on running LVM in a clustered environment.
- *Global File System: Configuration and Administration* — Provides information about installing, configuring, and maintaining Red Hat GFS (Red Hat Global File System).
- *Global File System 2: Configuration and Administration* — Provides information about installing, configuring, and maintaining Red Hat GFS2 (Red Hat Global File System 2).
- *Using Device-Mapper Multipath* — Provides information about using the Device-Mapper Multipath feature of Red Hat Enterprise Linux 5.
- *Using GNBD with Global File System* — Provides an overview on using Global Network Block Device (GNBD) with Red Hat GFS.
- *Linux Virtual Server Administration* — Provides information on configuring high-performance systems and services with the Linux Virtual Server (LVS).
- *Red Hat Cluster Suite Release Notes* — Provides information about the current release of Red Hat Cluster Suite.

Red Hat Cluster Suite documentation and other Red Hat documents are available in HTML, PDF, and RPM versions on the Red Hat Enterprise Linux Documentation CD and https://access.redhat.com/site/documentation/en-US/.

1. Document Conventions

This manual uses several conventions to highlight certain words and phrases and draw attention to specific pieces of information.

1.1. Typographic Conventions

Four typographic conventions are used to call attention to specific words and phrases. These conventions, and the circumstances they apply to, are as follows.

Mono-spaced Bold

Used to highlight system input, including shell commands, file names and paths. Also used to highlight keys and key combinations. For example:

> To see the contents of the file **my_next_bestselling_novel** in your current working directory, enter the **cat my_next_bestselling_novel** command at the shell prompt and press **Enter** to execute the command.

The above includes a file name, a shell command and a key, all presented in mono-spaced bold and all distinguishable thanks to context.

Key combinations can be distinguished from an individual key by the plus sign that connects each part of a key combination. For example:

> Press **Enter** to execute the command.
>
> Press **Ctrl**+**Alt**+**F2** to switch to a virtual terminal.

The first example highlights a particular key to press. The second example highlights a key combination: a set of three keys pressed simultaneously.

If source code is discussed, class names, methods, functions, variable names and returned values mentioned within a paragraph will be presented as above, in **mono-spaced bold**. For example:

> File-related classes include **filesystem** for file systems, **file** for files, and **dir** for directories. Each class has its own associated set of permissions.

Proportional Bold

This denotes words or phrases encountered on a system, including application names; dialog-box text; labeled buttons; check-box and radio-button labels; menu titles and submenu titles. For example:

> Choose **System** → **Preferences** → **Mouse** from the main menu bar to launch **Mouse Preferences**. In the **Buttons** tab, select the **Left-handed mouse** check box and click **Close** to switch the primary mouse button from the left to the right (making the mouse suitable for use in the left hand).
>
> To insert a special character into a **gedit** file, choose **Applications** → **Accessories** → **Character Map** from the main menu bar. Next, choose **Search** → **Find...** from the **Character Map** menu bar, type the name of the character in the **Search** field and click **Next**. The character you sought will be highlighted in the **Character Table**. Double-click this highlighted character to place it in the **Text to copy** field and then click the **Copy** button. Now switch back to your document and choose **Edit** → **Paste** from the **gedit** menu bar.

The above text includes application names; system-wide menu names and items; application-specific menu names; and buttons and text found within a GUI interface, all presented in proportional bold and all distinguishable by context.

Mono-spaced Bold Italic or ***Proportional Bold Italic***

Whether mono-spaced bold or proportional bold, the addition of italics indicates replaceable or variable text. Italics denotes text you do not input literally or displayed text that changes depending on circumstance. For example:

To connect to a remote machine using ssh, type **ssh *username*@*domain.name*** at a shell prompt. If the remote machine is **example.com** and your username on that machine is john, type **ssh john@example.com**.

The **mount -o remount *file-system*** command remounts the named file system. For example, to remount the **/home** file system, the command is **mount -o remount /home**.

To see the version of a currently installed package, use the **rpm -q *package*** command. It will return a result as follows: ***package-version-release***.

Note the words in bold italics above: username, domain.name, file-system, package, version and release. Each word is a placeholder, either for text you enter when issuing a command or for text displayed by the system.

Aside from standard usage for presenting the title of a work, italics denotes the first use of a new and important term. For example:

Publican is a *DocBook* publishing system.

1.2. Pull-quote Conventions

Terminal output and source code listings are set off visually from the surrounding text.

Output sent to a terminal is set in **mono-spaced roman** and presented thus:

```
books        Desktop   documentation  drafts  mss    photos   stuff  svn
books_tests  Desktop1  downloads      images  notes  scripts  svgs
```

Source-code listings are also set in **mono-spaced roman** but add syntax highlighting as follows:

```
static int kvm_vm_ioctl_deassign_device(struct kvm *kvm,
                 struct kvm_assigned_pci_dev *assigned_dev)
{
         int r = 0;
         struct kvm_assigned_dev_kernel *match;

         mutex_lock(&kvm->lock);

         match = kvm_find_assigned_dev(&kvm->arch.assigned_dev_head,
                                       assigned_dev->assigned_dev_id);
         if (!match) {
                 printk(KERN_INFO "%s: device hasn't been assigned before, "
                   "so cannot be deassigned\n", __func__);
                 r = -EINVAL;
                 goto out;
         }

         kvm_deassign_device(kvm, match);

         kvm_free_assigned_device(kvm, match);

out:
         mutex_unlock(&kvm->lock);
         return r;
}
```

1.3. Notes and Warnings

Finally, we use three visual styles to draw attention to information that might otherwise be overlooked.

Note

Notes are tips, shortcuts or alternative approaches to the task at hand. Ignoring a note should have no negative consequences, but you might miss out on a trick that makes your life easier.

Important

Important boxes detail things that are easily missed: configuration changes that only apply to the current session, or services that need restarting before an update will apply. Ignoring a box labeled “Important” will not cause data loss but may cause irritation and frustration.

Warning

Warnings should not be ignored. Ignoring warnings will most likely cause data loss.

2. Feedback

If you spot a typo, or if you have thought of a way to make this manual better, we would love to hear from you. Please submit a report in Bugzilla (http://bugzilla.redhat.com/bugzilla/). File the bug against the product **Red Hat Enterprise Linux 5** and against the component **Documentation-cluster**.

Be sure to mention the manual identifier:

```
Cluster_Administration(EN)-5 (2014-6-30T15:52)
```

By mentioning this manual's identifier, we know exactly which version of the guide you have.

If you have a suggestion for improving the documentation, try to be as specific as possible. If you have found an error, please include the section number and some of the surrounding text so we can find it easily.

Chapter 1. Red Hat Cluster Configuration and Management Overview

Red Hat Cluster allows you to connect a group of computers (called *nodes* or *members*) to work together as a cluster. It provides a wide variety of ways to configure hardware and software to suit your clustering needs (for example, a cluster for sharing files on a GFS file system or a cluster with high-availability service failover). This book provides information about how to use configuration tools to configure your cluster and provides considerations to take into account before deploying a Red Hat Cluster. To ensure that your deployment of Red Hat Cluster fully meets your needs and can be supported, consult with an authorized Red Hat representative before you deploy it.

1.1. Configuration Basics

To set up a cluster, you must connect the nodes to certain cluster hardware and configure the nodes into the cluster environment. This chapter provides an overview of cluster configuration and management, and tools available for configuring and managing a Red Hat Cluster.

Note

For information on best practices for deploying and upgrading Red Hat Enterprise Linux 5 Advanced Platform (Clustering and GFS/GFS2), refer to the article "Red Hat Enterprise Linux Cluster, High Availability, and GFS Deployment Best Practices" on Red Hat Customer Portal at https://access.redhat.com/site/articles/40051.

Configuring and managing a Red Hat Cluster consists of the following basic steps:

1. Setting up hardware. Refer to Section 1.1.1, “Setting Up Hardware”.
2. Installing Red Hat Cluster software. Refer to Section 1.1.2, “Installing Red Hat Cluster software”.
3. Configuring Red Hat Cluster Software. Refer to Section 1.1.3, “Configuring Red Hat Cluster Software”.

1.1.1. Setting Up Hardware

Setting up hardware consists of connecting cluster nodes to other hardware required to run a Red Hat Cluster. The amount and type of hardware varies according to the purpose and availability requirements of the cluster. Typically, an enterprise-level cluster requires the following type of hardware (refer to Figure 1.1, “Red Hat Cluster Hardware Overview”).For considerations about hardware and other cluster configuration concerns, refer to "Before Configuring a Red Hat Cluster" or check with an authorized Red Hat representative.

- Cluster nodes — Computers that are capable of running Red Hat Enterprise Linux 5 software, with at least 1GB of RAM. The maximum number of nodes supported in a Red Hat Cluster is 16.
- Ethernet switch or hub for public network — This is required for client access to the cluster.
- Ethernet switch or hub for private network — This is required for communication among the cluster nodes and other cluster hardware such as network power switches and Fibre Channel switches.
- Network power switch — A network power switch is recommended to perform fencing in an enterprise-level cluster.

- Fibre Channel switch — A Fibre Channel switch provides access to Fibre Channel storage. Other options are available for storage according to the type of storage interface; for example, iSCSI or GNBD. A Fibre Channel switch can be configured to perform fencing.
- Storage — Some type of storage is required for a cluster. The type required depends on the purpose of the cluster.

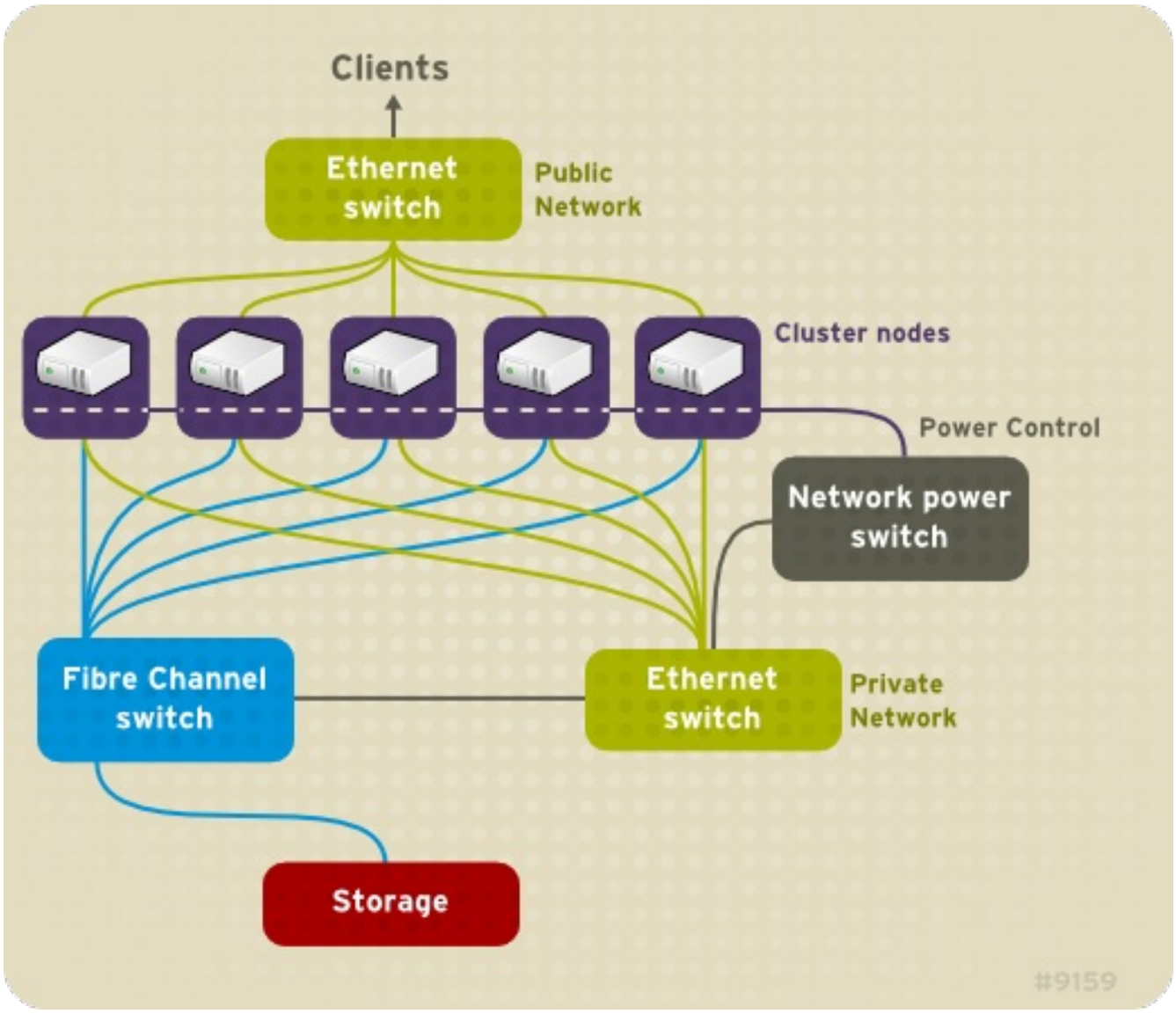


Figure 1.1. Red Hat Cluster Hardware Overview

1.1.2. Installing Red Hat Cluster software

To install Red Hat Cluster software, you must have entitlements for the software. If you are using the **Conga** configuration GUI, you can let it install the cluster software. If you are using other tools to configure the cluster, secure and install the software as you would with Red Hat Enterprise Linux software.

1.1.2.1. Upgrading the Cluster Software

It is possible to upgrade the cluster software on a given minor release of Red Hat Enterprise Linux without taking the cluster out of production. Doing so requires disabling the cluster software on one host at a time, upgrading the software, and restarting the cluster software on that host.

1. Shut down all cluster services on a single cluster node. For instructions on stopping cluster software on a node, refer to Section 6.1, “Starting and Stopping the Cluster Software”. It may be desirable to manually relocate cluster-managed services and virtual machines off of the host prior to stopping rgmanager.
2. Execute the yum update command to install the new RPMs. For example:

```
yum update -y openais cman rgmanager lvm2-cluster gfs2-utils
```

3. Reboot the cluster node or restart the cluster services manually. For instructions on starting cluster software on a node, refer to Section 6.1, “Starting and Stopping the Cluster Software”.

1.1.3. Configuring Red Hat Cluster Software

Configuring Red Hat Cluster software consists of using configuration tools to specify the relationship among the cluster components. Figure 1.2, “Cluster Configuration Structure” shows an example of the hierarchical relationship among cluster nodes, high-availability services, and resources. The cluster nodes are connected to one or more fencing devices. Nodes can be grouped into a failover domain for a cluster service. The services comprise resources such as NFS exports, IP addresses, and shared GFS partitions.

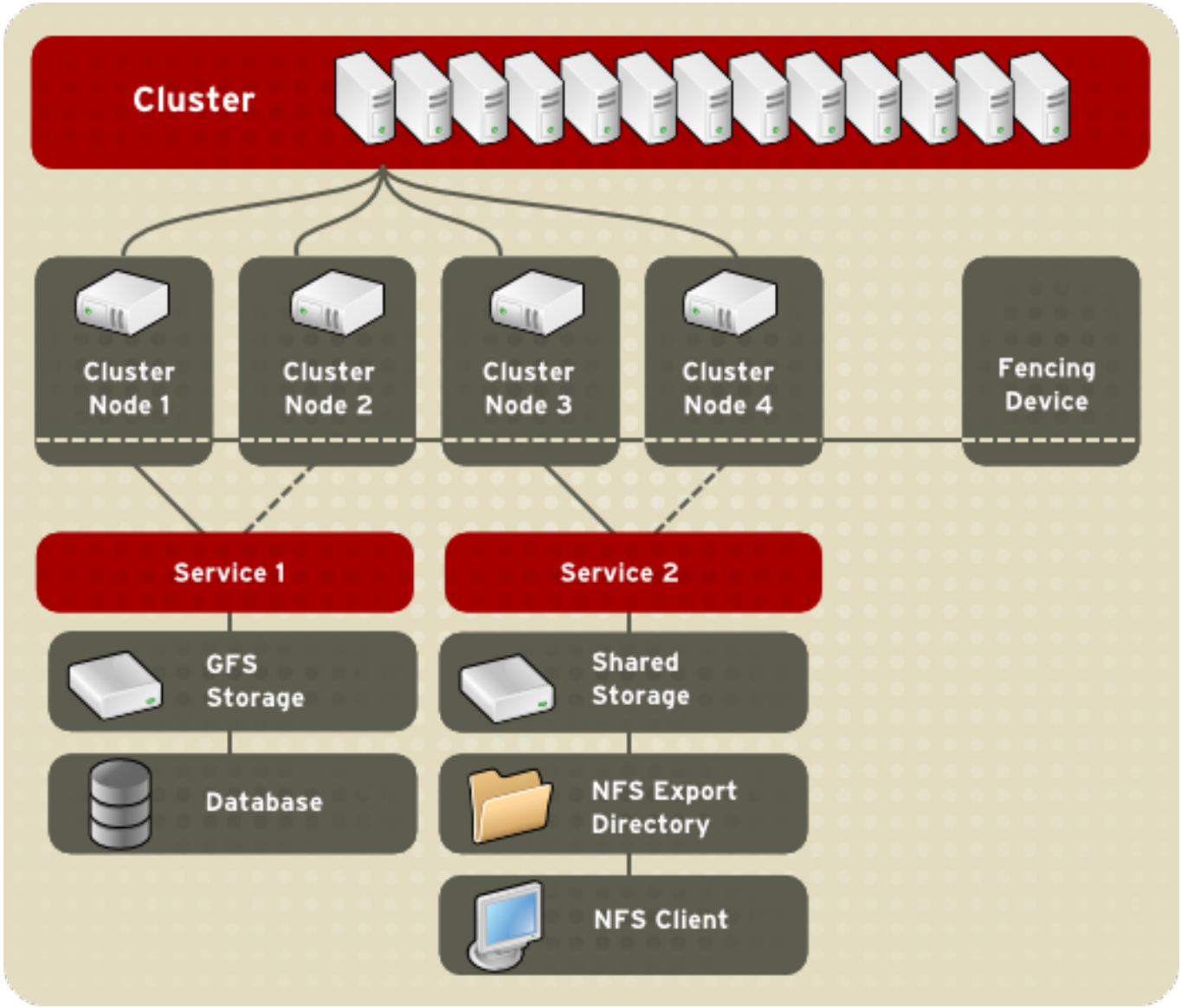


Figure 1.2. Cluster Configuration Structure

The following cluster configuration tools are available with Red Hat Cluster:

- **Conga** — This is a comprehensive user interface for installing, configuring, and managing Red Hat clusters, computers, and storage attached to clusters and computers.
- **system-config-cluster** — This is a user interface for configuring and managing a Red Hat cluster.
- Command line tools — This is a set of command line tools for configuring and managing a Red Hat cluster.

A brief overview of each configuration tool is provided in the following sections:

- Section 1.2, “Conga”
- Section 1.3, “**system-config-cluster** Cluster Administration GUI”
- Section 1.4, “Command Line Administration Tools”

In addition, information about using **Conga** and **system-config-cluster** is provided in subsequent chapters of this document. Information about the command line tools is available in the man pages for the tools.

1.2. Conga

Conga is an integrated set of software components that provides centralized configuration and management of Red Hat clusters and storage. **Conga** provides the following major features:

- One Web interface for managing cluster and storage
- Automated Deployment of Cluster Data and Supporting Packages
- Easy Integration with Existing Clusters
- No Need to Re-Authenticate
- Integration of Cluster Status and Logs
- Fine-Grained Control over User Permissions

The primary components in **Conga** are **luci** and **ricci**, which are separately installable. **luci** is a server that runs on one computer and communicates with multiple clusters and computers via **ricci**. **ricci** is an agent that runs on each computer (either a cluster member or a standalone computer) managed by **Conga**.

luci is accessible through a Web browser and provides three major functions that are accessible through the following tabs:

- **homebase** — Provides tools for adding and deleting computers, adding and deleting users, and configuring user privileges. Only a system administrator is allowed to access this tab.
- **cluster** — Provides tools for creating and configuring clusters. Each instance of **luci** lists clusters that have been set up with that **luci**. A system administrator can administer all clusters listed on this tab. Other users can administer only clusters that the user has permission to manage (granted by an administrator).
- **storage** — Provides tools for remote administration of storage. With the tools on this tab, you can manage storage on computers whether they belong to a cluster or not.

To administer a cluster or storage, an administrator adds (or *registers*) a cluster or a computer to a **luci** server. When a cluster or a computer is registered with **luci**, the FQDN hostname or IP address of each computer is stored in a **luci** database.

You can populate the database of one **luci** instance from another **luci**instance. That capability provides a means of replicating a **luci** server instance and provides an efficient upgrade and testing path. When you install an instance of **luci**, its database is empty. However, you can import part or all of a **luci** database from an existing **luci** server when deploying a new **luci** server.

Each **luci** instance has one user at initial installation — admin. Only the admin user may add systems to a **luci** server. Also, the admin user can create additional user accounts and determine which users are allowed to access clusters and computers registered in the **luci** database. It is possible to import users as a batch operation in a new **luci** server, just as it is possible to import clusters and computers.

When a computer is added to a **luci** server to be administered, authentication is done once. No authentication is necessary from then on (unless the certificate used is revoked by a CA). After that, you can remotely configure and manage clusters and storage through the **luci** user interface. **luci** and **ricci** communicate with each other via XML.

The following figures show sample displays of the three major **luci** tabs: **homebase**, **cluster**, and **storage**.

For more information about **Conga**, refer to Chapter 3, *Configuring Red Hat Cluster With **Conga***, Chapter 4, *Managing Red Hat Cluster With **Conga***, and the online help available with the **luci** server.

Figure 1.3. luci homebase Tab

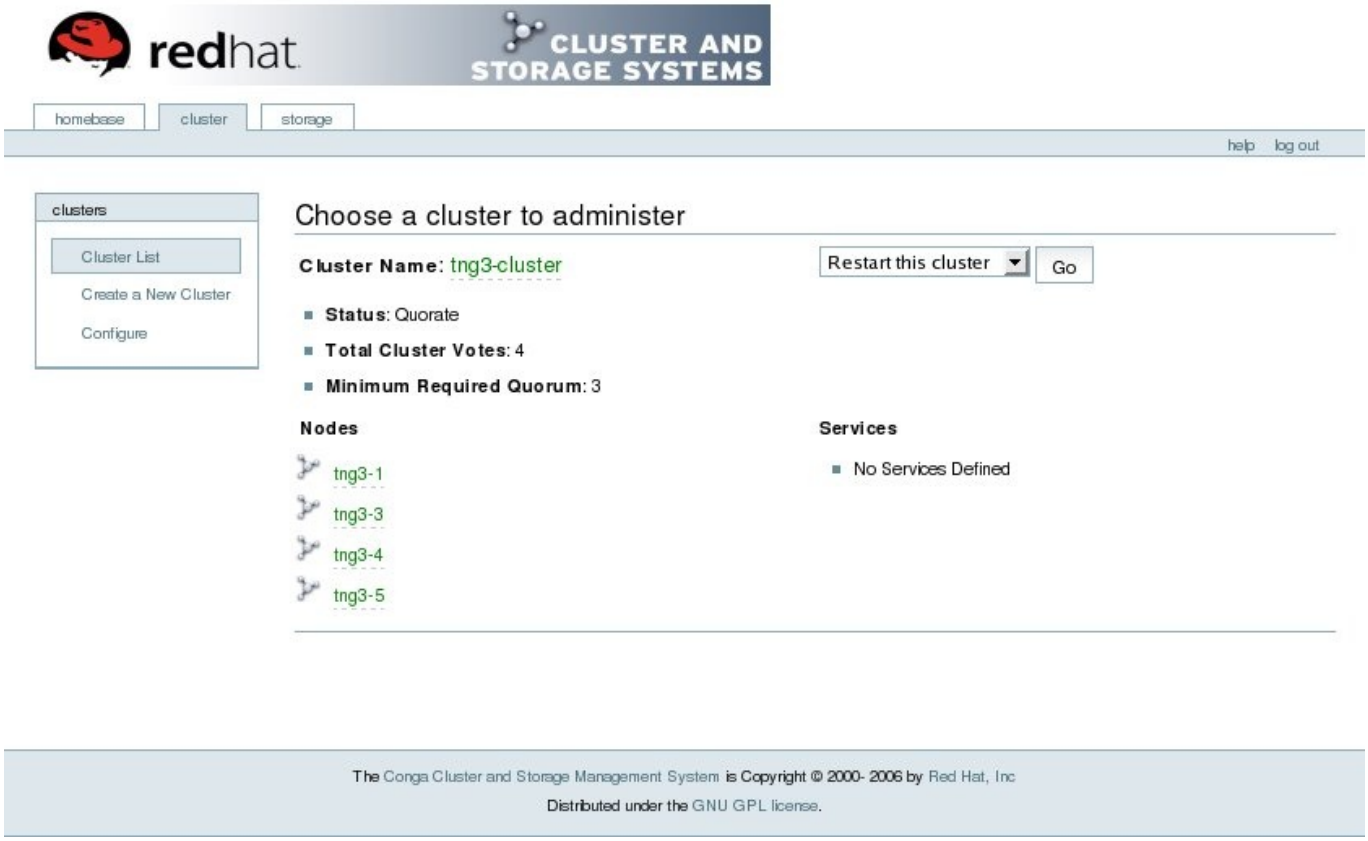


Figure 1.4. luci cluster Tab

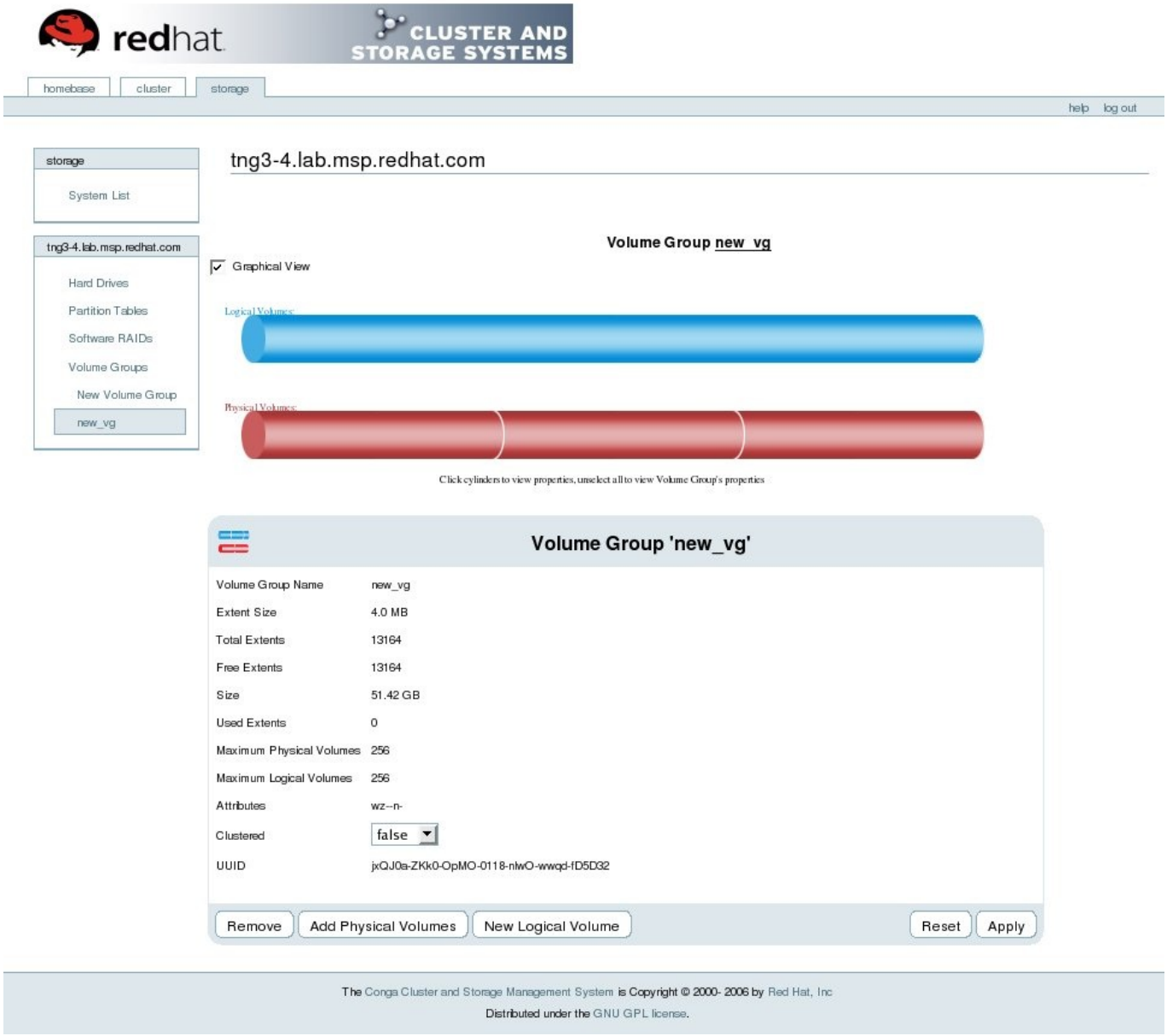


Figure 1.5. luci storage Tab

1.3. `system-config-cluster` Cluster Administration GUI

This section provides an overview of the cluster administration graphical user interface (GUI) available with Red Hat Cluster Suite — **`system-config-cluster`**. It is for use with the cluster infrastructure and the high-availability service management components. **`system-config-cluster`** consists of two major functions: the **Cluster Configuration Tool** and the **Cluster Status Tool**. The **Cluster Configuration Tool** provides the capability to create, edit, and propagate the cluster configuration file (**`/etc/cluster/cluster.conf`**). The **Cluster Status Tool** provides the capability to manage high-availability services. The following sections summarize those functions.

Note

While **system-config-cluster** provides several convenient tools for configuring and managing a Red Hat Cluster, the newer, more comprehensive tool, **Conga**, provides more convenience and flexibility than **system-config-cluster**.

1.3.1. Cluster Configuration Tool

You can access the **Cluster Configuration Tool** (Figure 1.6, "Cluster Configuration Tool") through the **Cluster Configuration** tab in the Cluster Administration GUI.

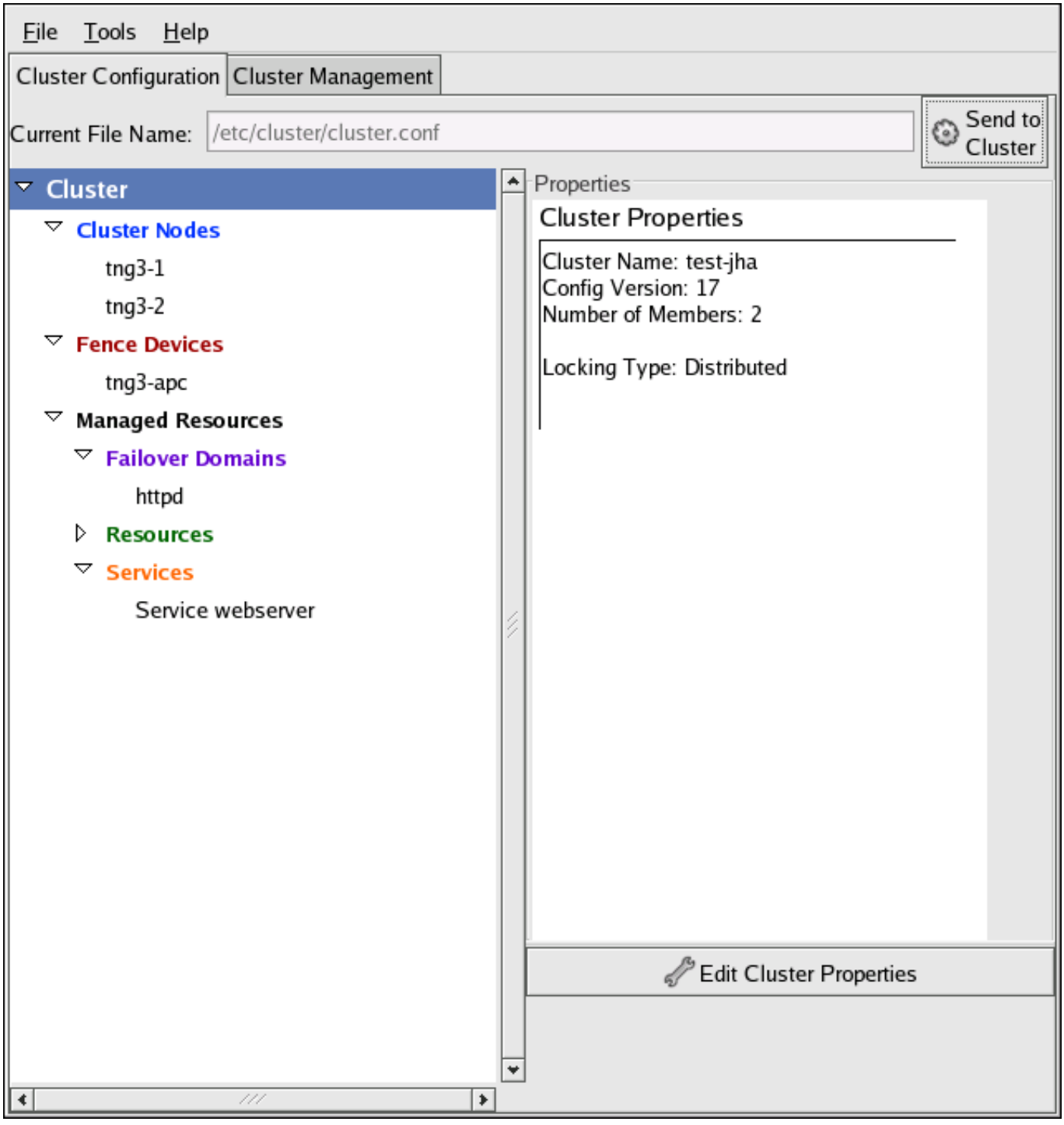


Figure 1.6. Cluster Configuration Tool

The **Cluster Configuration Tool** represents cluster configuration components in the configuration file (**/etc/cluster/cluster.conf**) with a hierarchical graphical display in the left panel. A triangle icon to the left of a component name indicates that the component has one or more subordinate components assigned to it. Clicking the triangle icon expands and collapses the portion of the tree below a component. The components displayed in the GUI are summarized as follows:

- **Cluster Nodes** — Displays cluster nodes. Nodes are represented by name as subordinate elements under **Cluster Nodes**. Using configuration buttons at the bottom of the right frame (below **Properties**), you can add nodes, delete nodes, edit node properties, and configure fencing methods for each node.
- **Fence Devices** — Displays fence devices. Fence devices are represented as subordinate elements under **Fence Devices**. Using configuration buttons at the bottom of the right frame (below **Properties**), you can add fence devices, delete fence devices, and edit fence-device properties. Fence devices must be defined before you can configure fencing (with the **Manage Fencing For This Node** button) for each node.
- **Managed Resources** — Displays failover domains, resources, and services.
  - **Failover Domains** — For configuring one or more subsets of cluster nodes used to run a high-availability service in the event of a node failure. Failover domains are represented as subordinate elements under **Failover Domains**. Using configuration buttons at the bottom of the right frame (below **Properties**), you can create failover domains (when **Failover Domains** is selected) or edit failover domain properties (when a failover domain is selected).
  - **Resources** — For configuring shared resources to be used by high-availability services. Shared resources consist of file systems, IP addresses, NFS mounts and exports, and user-created scripts that are available to any high-availability service in the cluster. Resources are represented as subordinate elements under **Resources**. Using configuration buttons at the bottom of the right frame (below **Properties**), you can create resources (when **Resources** is selected) or edit resource properties (when a resource is selected).

    Note

    The **Cluster Configuration Tool** provides the capability to configure private resources, also. A private resource is a resource that is configured for use with only one service. You can configure a private resource within a **Service** component in the GUI.

  - **Services** — For creating and configuring high-availability services. A service is configured by assigning resources (shared or private), assigning a failover domain, and defining a recovery policy for the service. Services are represented as subordinate elements under **Services**. Using configuration buttons at the bottom of the right frame (below **Properties**), you can create services (when **Services** is selected) or edit service properties (when a service is selected).

1.3.2. Cluster Status Tool

You can access the **Cluster Status Tool** (Figure 1.7, “Cluster Status Tool”) through the **Cluster Management** tab in Cluster Administration GUI.

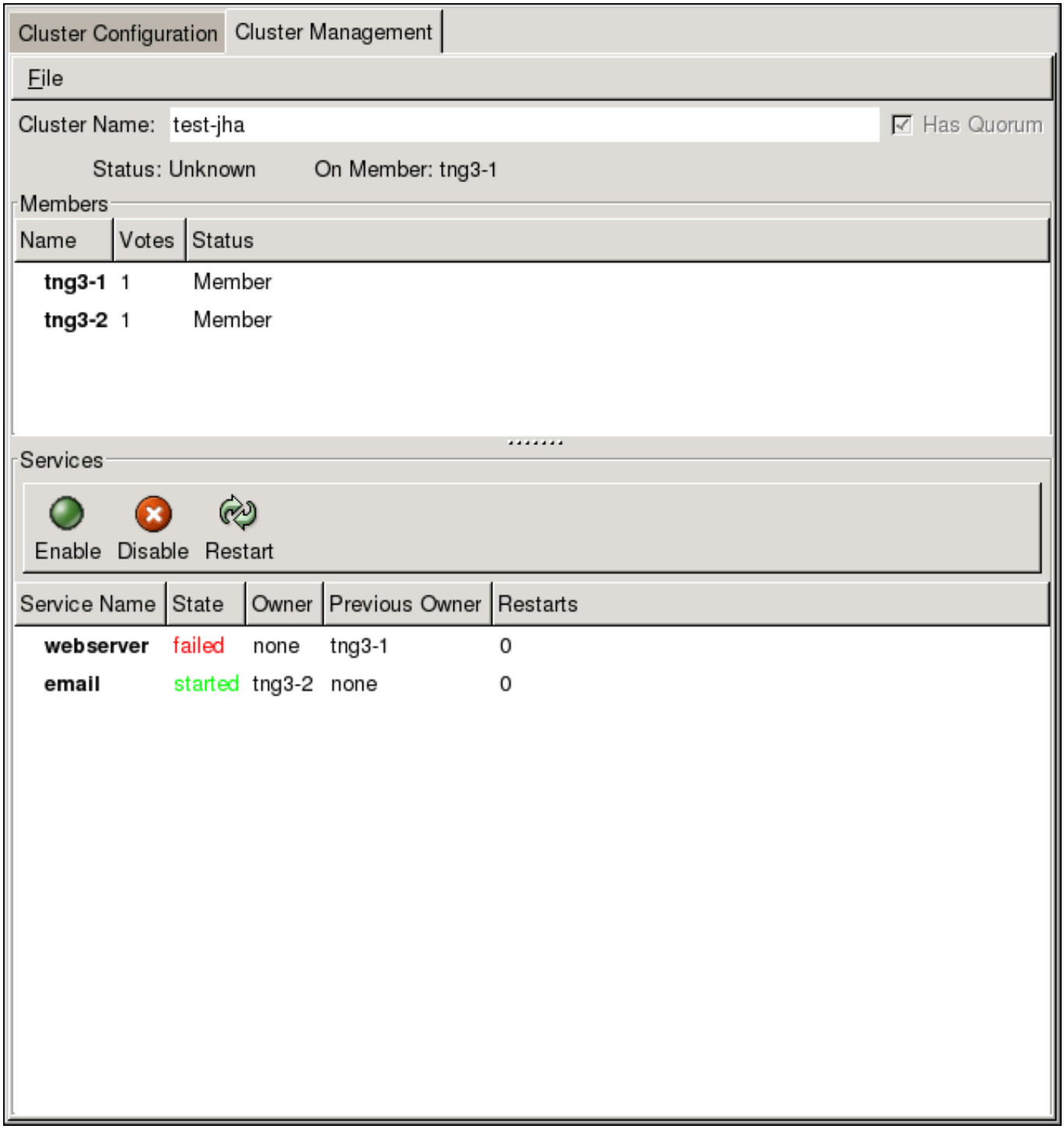


Figure 1.7. Cluster Status Tool

The nodes and services displayed in the **Cluster Status Tool** are determined by the cluster configuration file (**/etc/cluster/cluster.conf**). You can use the **Cluster Status Tool** to enable, disable, restart, or relocate a high-availability service.

1.4. Command Line Administration Tools

In addition to **Conga** and the **system-config-cluster** Cluster Administration GUI, command line tools are available for administering the cluster infrastructure and the high-availability service management components. The command line tools are used by the Cluster Administration GUI and init scripts supplied by Red Hat. Table 1.1, “Command Line Tools” summarizes the command line tools.

Table 1.1. Command Line Tools

Command Line Tool	Used With	Purpose
ccs_tool — Cluster Configuration System Tool	Cluster Infrastructure	**ccs_tool** is a program for making online updates to the cluster configuration file. It provides the capability to create and modify cluster infrastructure components (for example, creating a cluster, adding and removing a node). For more information about this tool, refer to the ccs_tool(8) man page.
cman_tool — Cluster Management Tool	Cluster Infrastructure	**cman_tool** is a program that manages the CMAN cluster manager. It provides the capability to join a cluster, leave a cluster, kill a node, or change the expected quorum votes of a node in a cluster. For more information about this tool, refer to the cman_tool(8) man page.
fence_tool — Fence Tool	Cluster Infrastructure	**fence_tool** is a program used to join or leave the default fence domain. Specifically, it starts the fence daemon (**fenced**) to join the domain and kills **fenced** to leave the domain. For more information about this tool, refer to the fence_tool(8) man page.
clustat — Cluster Status Utility	High-availability Service Management Components	The **clustat** command displays the status of the cluster. It shows membership information, quorum view, and the state of all configured user services. For more information about this tool, refer to the clustat(8) man page.
clusvcadm — Cluster User Service Administration Utility	High-availability Service Management Components	The **clusvcadm** command allows you to enable, disable, relocate, and restart high-availability services in a cluster. For more information about this tool, refer to the clusvcadm(8) man page.

Chapter 2. Before Configuring a Red Hat Cluster

This chapter describes tasks to perform and considerations to make before installing and configuring a Red Hat Cluster, and consists of the following sections.

Important

Make sure that your deployment of Red Hat Cluster Suite meets your needs and can be supported. Consult with an authorized Red Hat representative to verify Cluster Suite and GFS configuration prior to deployment. In addition, allow time for a configuration burn-in period to test failure modes.

- Section 2.1, “General Configuration Considerations”
- Section 2.2, “Compatible Hardware”
- Section 2.3, “Enabling IP Ports”
- Section 2.4, “Configuring ACPI For Use with Integrated Fence Devices”
- Section 2.6, “Configuring max_luns”
- Section 2.7, “Considerations for Using Quorum Disk”
- Section 2.8, “Red Hat Cluster Suite and SELinux”
- Section 2.9, “Multicast Addresses”
- Section 2.10, “Configuring the iptables Firewall to Allow Cluster Components”
- Section 2.11, “Considerations for Using **Conga**”
- Section 2.12, “Configuring Virtual Machines in a Clustered Environment”

2.1. General Configuration Considerations

You can configure a Red Hat Cluster in a variety of ways to suit your needs. Take into account the following general considerations when you plan, configure, and implement your Red Hat Cluster.

Number of cluster nodes supported

The maximum number of nodes supported in a Red Hat Cluster is 16.

GFS/GFS2

Although a GFS/GFS2 file system can be implemented in a standalone system or as part of a cluster configuration, for the RHEL 5.5 release and later, Red Hat does not support the use of GFS/GFS2 as a single-node file system. Red Hat does support a number of high-performance single-node file systems that are optimized for single node, and thus have generally lower overhead than a cluster file system. Red Hat recommends using those file systems in preference to GFS/GFS2 in cases where only a single node needs to mount the file system. Red Hat will continue to support single-node GFS/GFS2 file systems for existing customers.

When you configure a GFS/GFS2 file system as a cluster file system, you must ensure that all nodes in the cluster have access to the shared file system. Asymmetric cluster

configurations in which some nodes have access to the file system and others do not are not supported.This does not require that all nodes actually mount the GFS/GFS2 file system itself.

No-single-point-of-failure hardware configuration

Clusters can include a dual-controller RAID array, multiple bonded network channels, multiple paths between cluster members and storage, and redundant un-interruptible power supply (UPS) systems to ensure that no single failure results in application down time or loss of data.

Alternatively, a low-cost cluster can be set up to provide less availability than a no-single-point-of-failure cluster. For example, you can set up a cluster with a single-controller RAID array and only a single Ethernet channel.

Certain low-cost alternatives, such as host RAID controllers, software RAID without cluster support, and multi-initiator parallel SCSI configurations are not compatible or appropriate for use as shared cluster storage.

Data integrity assurance

To ensure data integrity, only one node can run a cluster service and access cluster-service data at a time. The use of power switches in the cluster hardware configuration enables a node to power-cycle another node before restarting that node's HA services during a failover process. This prevents two nodes from simultaneously accessing the same data and corrupting it. It is strongly recommended that *fence devices* (hardware or software solutions that remotely power, shutdown, and reboot cluster nodes) are used to guarantee data integrity under all failure conditions. Watchdog timers provide an alternative way to to ensure correct operation of HA service failover.

Ethernet channel bonding

Cluster quorum and node health is determined by communication of messages among cluster nodes via Ethernet. In addition, cluster nodes use Ethernet for a variety of other critical cluster functions (for example, fencing). With Ethernet channel bonding, multiple Ethernet interfaces are configured to behave as one, reducing the risk of a single-point-of-failure in the typical switched Ethernet connection among cluster nodes and other cluster hardware.

Red Hat Enterprise Linux 5 supports bonding mode 1 only. It is recommended that you wire each node's slaves to the switches in a consistent manner, with each node's primary device wired to switch 1 and each node's backup device wired to switch 2.

2.2. Compatible Hardware

Before configuring Red Hat Cluster software, make sure that your cluster uses appropriate hardware (for example, supported fence devices, storage devices, and Fibre Channel switches). Refer to the Red Hat Hardware Catalog at https://hardware.redhat.com/ for the most current hardware compatibility information.

2.3. Enabling IP Ports

Before deploying a Red Hat Cluster, you must enable certain IP ports on the cluster nodes and on computers that run **luci** (the **Conga** user interface server). The following sections identify the IP ports to be enabled:

- Section 2.3.1, “Enabling IP Ports on Cluster Nodes”
- Section 2.3.2, “Enabling IP Ports on Computers That Run **luci**”

2.3.1. Enabling IP Ports on Cluster Nodes

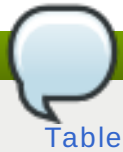
To allow Red Hat Cluster nodes to communicate with each other, you must enable the IP ports assigned to certain Red Hat Cluster components. Table 2.1, “Enabled IP Ports on Red Hat Cluster Nodes” lists the IP port numbers, their respective protocols, and the components to which the port numbers are assigned. At each cluster node, enable IP ports according to Table 2.1, “Enabled IP Ports on Red Hat Cluster Nodes”.

Note

IPV6 is not supported for Cluster Suite in Red Hat Enterprise Linux 5.

Table 2.1. Enabled IP Ports on Red Hat Cluster Nodes

IP Port Number	Protocol	Component
5404, 5405	UDP	**cman** (Cluster Manager)
11111	TCP	**ricci** (part of **Conga** remote agent)
14567	TCP	**gnbd** (Global Network Block Device)
16851	TCP	**modclusterd** (part of **Conga** remote agent)
21064	TCP	**dlm** (Distributed Lock Manager)
50006, 50008, 50009	TCP	**ccsd** (Cluster Configuration System daemon)
50007	UDP	**ccsd** (Cluster Configuration System daemon)



Note

Table 2.1, “Enabled IP Ports on Red Hat Cluster Nodes” shows no IP ports to enable for **rgmanager**. For Red Hat Enterprise Linux 5.1 and later, **rgmanager** does not use TCP or UDP sockets.

2.3.2. Enabling IP Ports on Computers That Run luci

To allow client computers to communicate with a computer that runs **luci** (the **Conga** user interface server), and to allow a computer that runs **luci** to communicate with **ricci** in the cluster nodes, you must enable the IP ports assigned to **luci** and **ricci**. Table 2.1, “Enabled IP Ports on Red Hat Cluster Nodes” lists the IP port numbers, their respective protocols, and the components to which the port numbers are assigned. At each computer that runs **luci**, enable IP ports according to Table 2.2, “Enabled IP Ports on a Computer That Runs luci”.

Note

If a cluster node is running **luci**, port 11111 should already have been enabled.

Table 2.2. Enabled IP Ports on a Computer That Runs luci

IP Port Number	Protocol	Component
8084	TCP	**luci** (**Conga** user interface server)
11111	TCP	**ricci** (**Conga** remote agent)

If your server infrastructure incorporates more than one network and you want to access **luci** from the internal network only, you can configure the **stunnel** component to listen on one IP address only by editing the **LUCI_HTTPS_PORT** parameter in the **/etc/sysconfig/luci** file as follows:

```
LUCI_HTTPS_PORT=10.10.10.10:8084
```

2.4. Configuring ACPI For Use with Integrated Fence Devices

If your cluster uses integrated fence devices, you must configure ACPI (Advanced Configuration and Power Interface) to ensure immediate and complete fencing.

Note

For the most current information about integrated fence devices supported by Red Hat Cluster Suite, refer to http://www.redhat.com/cluster_suite/hardware/.

If a cluster node is configured to be fenced by an integrated fence device, disable ACPI Soft-Off for that node. Disabling ACPI Soft-Off allows an integrated fence device to turn off a node immediately and completely rather than attempting a clean shutdown (for example, **shutdown -h now**). Otherwise, if ACPI Soft-Off is enabled, an integrated fence device can take four or more seconds to turn off a node (refer to note that follows). In addition, if ACPI Soft-Off is enabled and a node panics or freezes during shutdown, an integrated fence device may not be able to turn off the node. Under those circumstances, fencing is delayed or unsuccessful. Consequently, when a node is fenced with an integrated fence device and ACPI Soft-Off is enabled, a cluster recovers slowly or requires administrative intervention to recover.

Note

The amount of time required to fence a node depends on the integrated fence device used. Some integrated fence devices perform the equivalent of pressing and holding the power button; therefore, the fence device turns off the node in four to five seconds. Other integrated fence devices perform the equivalent of pressing the power button momentarily, relying on the operating system to turn off the node; therefore, the fence device turns off the node in a time span much longer than four to five seconds.

To disable ACPI Soft-Off, use **chkconfig** management and verify that the node turns off immediately when fenced. The preferred way to disable ACPI Soft-Off is with **chkconfig** management: however, if that method is not satisfactory for your cluster, you can disable ACPI Soft-Off with one of the following alternate methods:

- Changing the BIOS setting to "instant-off" or an equivalent setting that turns off the node without delay

Note

Disabling ACPI Soft-Off with the BIOS may not be possible with some computers.

- Appending **acpi=off** to the kernel boot command line of the **/boot/grub/grub.conf** file

Important

This method completely disables ACPI; some computers do not boot correctly if ACPI is completely disabled. Use this method *only* if the other methods are not effective for your cluster.

The following sections provide procedures for the preferred method and alternate methods of disabling ACPI Soft-Off:

- Section 2.4.1, "Disabling ACPI Soft-Off with **chkconfig** Management" — Preferred method
- Section 2.4.2, "Disabling ACPI Soft-Off with the BIOS" — First alternate method
- Section 2.4.3, "Disabling ACPI Completely in the **grub.conf** File" — Second alternate method

2.4.1. Disabling ACPI Soft-Off with chkconfig Management

You can use **chkconfig** management to disable ACPI Soft-Off either by removing the ACPI daemon (**acpid**) from **chkconfig** management or by turning off **acpid**.

Note

This is the preferred method of disabling ACPI Soft-Off.

Disable ACPI Soft-Off with **chkconfig** management at each cluster node as follows:

1. Run either of the following commands:
   - **chkconfig --del acpid** — This command removes **acpid** from **chkconfig** management.

     — OR —

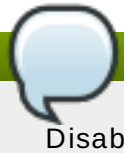
   - **chkconfig --level 2345 acpid off** — This command turns off **acpid**.
2. Reboot the node.
3. When the cluster is configured and running, verify that the node turns off immediately when fenced.

Note

You can fence the node with the **fence_node** command or **Conga**.

2.4.2. Disabling ACPI Soft-Off with the BIOS

The preferred method of disabling ACPI Soft-Off is with **chkconfig** management (Section 2.4.1, "Disabling ACPI Soft-Off with **chkconfig** Management"). However, if the preferred method is not effective for your cluster, follow the procedure in this section.



Note

Disabling ACPI Soft-Off with the BIOS may not be possible with some computers.

You can disable ACPI Soft-Off by configuring the BIOS of each cluster node as follows:

1. Reboot the node and start the **BIOS CMOS Setup Utility** program.
2. Navigate to the **Power** menu (or equivalent power management menu).
3. At the **Power** menu, set the **Soft-Off by PWR-BTTN** function (or equivalent) to **Instant-Off** (or the equivalent setting that turns off the node via the power button without delay). Example 2.1, "**BIOS CMOS Setup Utility**: Soft-Off by PWR-BTTN set to Instant-Off" shows a **Power** menu with **ACPI Function** set to **Enabled** and **Soft-Off by PWR-BTTN** set to **Instant-Off**.

Note

The equivalents to **ACPI Function**, **Soft-Off by PWR-BTTN**, and **Instant-Off** may vary among computers. However, the objective of this procedure is to configure the BIOS so that the computer is turned off via the power button without delay.

4. Exit the **BIOS CMOS Setup Utility** program, saving the BIOS configuration.
5. When the cluster is configured and running, verify that the node turns off immediately when fenced.

Note

You can fence the node with the **fence_node** command or **Conga**.

Example 2.1. BIOS CMOS Setup Utility: Soft-Off by PWR-BTTN set to Instant-Off

```
+---------------------------------------------|-------------------+
|    ACPI Function             [Enabled]      |    Item Help      |
|    ACPI Suspend Type         [S1(POS)]      |-------------------|
|  x Run VGABIOS if S3 Resume   Auto          |   Menu Level   *  |
```

```
|    Suspend Mode              [Disabled]     |                    |
|    HDD Power Down            [Disabled]     |                    |
|    Soft-Off by PWR-BTTN      [Instant-Off   |                    |
|    CPU THRM-Throttling       [50.0%]        |                    |
|    Wake-Up by PCI card       [Enabled]      |                    |
|    Power On by Ring          [Enabled]      |                    |
|    Wake Up On LAN            [Enabled]      |                    |
|  x USB KB Wake-Up From S3     Disabled      |                    |
|    Resume by Alarm           [Disabled]     |                    |
|  x  Date(of Month) Alarm       0            |                    |
|  x  Time(hh:mm:ss) Alarm       0 :  0 :     |                    |
|    POWER ON Function         [BUTTON ONLY   |                    |
|  x KB Power ON Password       Enter         |                    |
|  x Hot Key Power ON           Ctrl-F1       |                    |
|                                             |                    |
|                                             |                    |
+---------------------------------------------|--------------------+
```

This example shows **ACPI Function** set to **Enabled**, and **Soft-Off by PWR-BTTN** set to **Instant-Off**.

2.4.3. Disabling ACPI Completely in the grub.conf File

The preferred method of disabling ACPI Soft-Off is with **chkconfig** management (Section 2.4.1, "Disabling ACPI Soft-Off with **chkconfig** Management"). If the preferred method is not effective for your cluster, you can disable ACPI Soft-Off with the BIOS power management (Section 2.4.2, "Disabling ACPI Soft-Off with the BIOS"). If neither of those methods is effective for your cluster, you can disable ACPI completely by appending **acpi=off** to the kernel boot command line in the **grub.conf** file.

Important

This method completely disables ACPI; some computers do not boot correctly if ACPI is completely disabled. Use this method *only* if the other methods are not effective for your cluster.

You can disable ACPI completely by editing the **grub.conf** file of each cluster node as follows:

1. Open **/boot/grub/grub.conf** with a text editor.
2. Append **acpi=off** to the kernel boot command line in **/boot/grub/grub.conf** (refer to Example 2.2, "Kernel Boot Command Line with **acpi=off** Appended to It").
3. Reboot the node.
4. When the cluster is configured and running, verify that the node turns off immediately when fenced.

Note

You can fence the node with the **fence_node** command or **Conga**.

Example 2.2. Kernel Boot Command Line with acpi=off Appended to It

```
# grub.conf generated by anaconda
#
# Note that you do not have to rerun grub after making changes to this file
# NOTICE:  You have a /boot partition.  This means that
#          all kernel and initrd paths are relative to /boot/, eg.
#          root (hd0,0)
#          kernel /vmlinuz-version ro root=/dev/VolGroup00/LogVol00
#          initrd /initrd-version.img
#boot=/dev/hda
default=0
timeout=5
serial --unit=0 --speed=115200
terminal --timeout=5 serial console
title Red Hat Enterprise Linux Server (2.6.18-36.el5)
        root (hd0,0)
        kernel /vmlinuz-2.6.18-36.el5 ro root=/dev/VolGroup00/LogVol00
console=ttyS0,115200n8 acpi=off
        initrd /initrd-2.6.18-36.el5.img
```

In this example, **acpi=off** has been appended to the kernel boot command line — the line starting with "kernel /vmlinuz-2.6.18-36.el5".

2.5. Considerations for Configuring HA Services

You can create a cluster to suit your needs for high availability by configuring HA (high-availability) services. The key component for HA service management in a Red Hat cluster, **rgmanager**, implements cold failover for off-the-shelf applications. In a Red Hat cluster, an application is configured with other cluster resources to form an HA service that can fail over from one cluster node to another with no apparent interruption to cluster clients. HA-service failover can occur if a cluster node fails or if a cluster system administrator moves the service from one cluster node to another (for example, for a planned outage of a cluster node).

To create an HA service, you must configure it in the cluster configuration file. An HA service comprises cluster *resources*. Cluster resources are building blocks that you create and manage in the cluster configuration file — for example, an IP address, an application initialization script, or a Red Hat GFS shared partition.

An HA service can run on only one cluster node at a time to maintain data integrity. You can specify failover priority in a failover domain. Specifying failover priority consists of assigning a priority level to each node in a failover domain. The priority level determines the failover order — determining which node that an HA service should fail over to. If you do not specify failover priority, an HA service can fail over to any node in its failover domain. Also, you can specify if an HA service is restricted to run only on nodes of its associated failover domain. (When associated with an unrestricted failover domain, an HA service can start on any cluster node in the event no member of the failover domain is available.)

Figure 2.1, “Web Server Cluster Service Example” shows an example of an HA service that is a web server named "content-webserver". It is running in cluster node B and is in a failover domain that consists of nodes A, B, and D. In addition, the failover domain is configured with a failover priority to fail over to node D before node A and to restrict failover to nodes only in that failover domain. The HA service comprises these cluster resources:

- IP address resource — IP address 10.10.10.201.
- An application resource named "httpd-content" — a web server application init script **/etc/init.d/httpd** (specifying **httpd**).
- A file system resource — Red Hat GFS named "gfs-content-webserver".

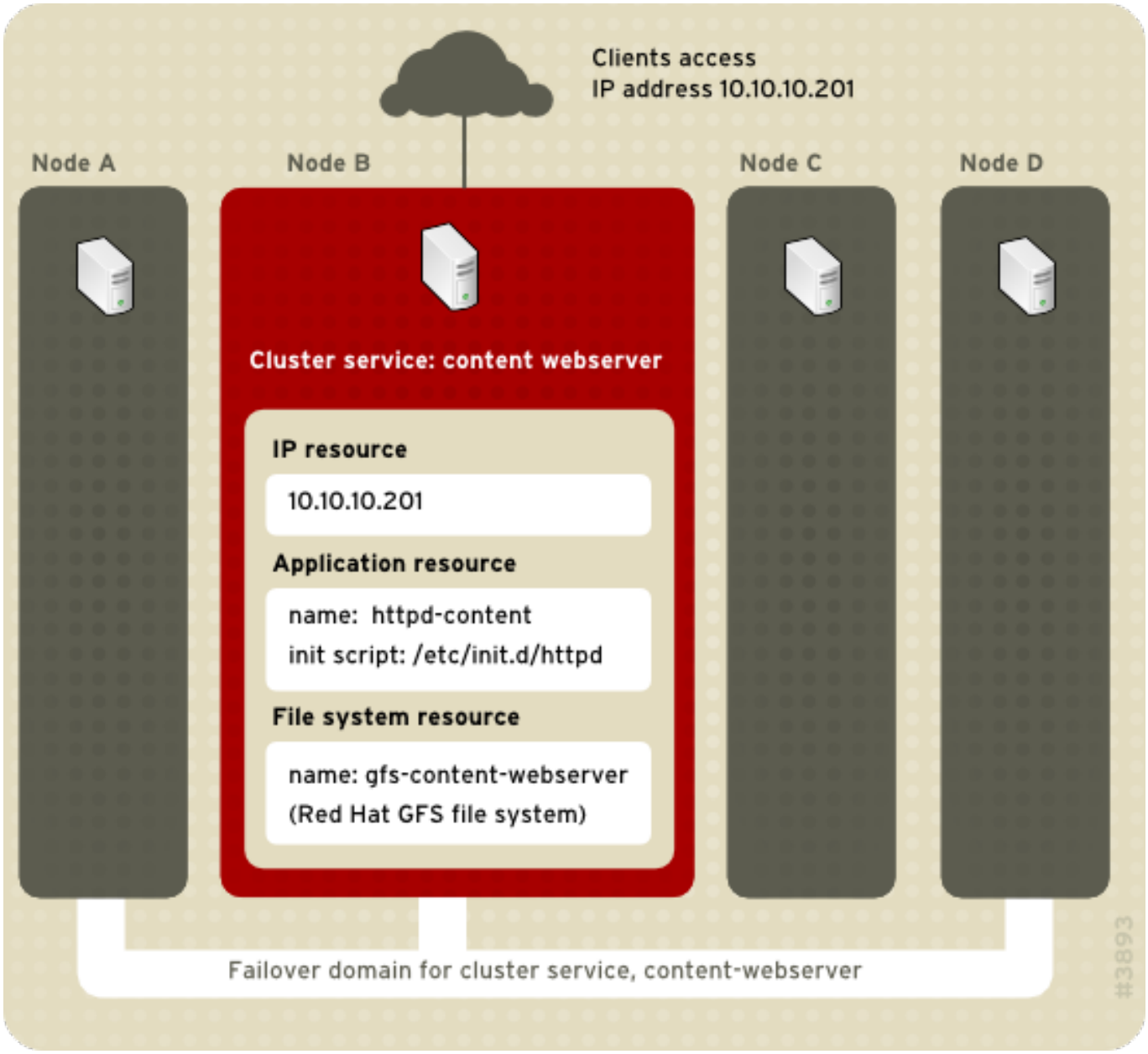


Figure 2.1. Web Server Cluster Service Example

Clients access the HA service through the IP address 10.10.10.201, enabling interaction with the web server application, httpd-content. The httpd-content application uses the gfs-content-webserver file system. If node B were to fail, the content-webserver HA service would fail over to node D. If node D were not available or also failed, the service would fail over to node A. Failover would occur with minimal service interruption to the cluster clients. For example, in an HTTP service, certain state information may be lost (like session data). The HA service would be accessible from another cluster node via the same IP address as it was before failover.

Note

For more information about HA services and failover domains, refer to *Red Hat Cluster Suite Overview*. For information about configuring failover domains, refer to Section 3.7, "Configuring a Failover Domain" (using **Conga**) or Section 5.6, "Configuring a Failover Domain" (using **system-config-cluster**).

An HA service is a group of cluster resources configured into a coherent entity that provides specialized services to clients. An HA service is represented as a resource tree in the cluster configuration file, **/etc/cluster/cluster.conf** (in each cluster node). In the cluster configuration file, each resource tree is an XML representation that specifies each resource, its attributes, and its relationship among other resources in the resource tree (parent, child, and sibling relationships).

Note

Because an HA service consists of resources organized into a hierarchical tree, a service is sometimes referred to as a *resource tree* or *resource group*. Both phrases are synonymous with *HA service*.

At the root of each resource tree is a special type of resource — a *service resource*. Other types of resources comprise the rest of a service, determining its characteristics. Configuring an HA service consists of creating a service resource, creating subordinate cluster resources, and organizing them into a coherent entity that conforms to hierarchical restrictions of the service.

Red Hat Cluster supports the following HA services:

- Apache
- Application (Script)
- LVM (HA LVM)
- MySQL
- NFS
- Open LDAP
- Oracle
- PostgreSQL 8
- Samba

Note

Red Hat Enterprise Linux 5 does not support running Clustered Samba in an active/active configuration, in which Samba serves the same shared file system from multiple nodes. Red Hat Enterprise Linux 5 does support running Samba in a cluster in active/passive mode, with failover from one node to the other nodes in a cluster. Note that if failover occurs, locking states are lost and active connections to Samba are severed so that the clients must reconnect.

- SAP
- Tomcat 5

There are two major considerations to take into account when configuring an HA service:

- The types of resources needed to create a service

- Parent, child, and sibling relationships among resources

The types of resources and the hierarchy of resources depend on the type of service you are configuring.

The types of cluster resources are listed in Appendix C, *HA Resource Parameters*. Information about parent, child, and sibling relationships among resources is described in Appendix D, *HA Resource Behavior*.

2.6. Configuring max_luns

It is *not* necessary to configure **max_luns** in Red Hat Enterprise Linux 5.

In Red Hat Enterprise Linux releases prior to Red Hat Enterprise Linux 5, if RAID storage in a cluster presents multiple LUNs, it is necessary to enable access to those LUNs by configuring **max_luns** (or **max_scsi_luns** for 2.4 kernels) in the **/etc/modprobe.conf** file of each node. In Red Hat Enterprise Linux 5, cluster nodes detect multiple LUNs without intervention required; it is *not* necessary to configure **max_luns** to detect multiple LUNs.

2.7. Considerations for Using Quorum Disk

Quorum Disk is a disk-based quorum daemon, **qdiskd**, that provides supplemental heuristics to determine node fitness. With heuristics you can determine factors that are important to the operation of the node in the event of a network partition. For example, in a four-node cluster with a 3:1 split, ordinarily, the three nodes automatically "win" because of the three-to-one majority. Under those circumstances, the one node is fenced. With **qdiskd** however, you can set up heuristics that allow the one node to win based on access to a critical resource (for example, a critical network path). If your cluster requires additional methods of determining node health, then you should configure **qdiskd** to meet those needs.

Note

Configuring **qdiskd** is not required unless you have special requirements for node health. An example of a special requirement is an "all-but-one" configuration. In an all-but-one configuration, **qdiskd** is configured to provide enough quorum votes to maintain quorum even though only one node is working.

Important

Overall, heuristics and other **qdiskd** parameters for your Red Hat Cluster depend on the site environment and special requirements needed. To understand the use of heuristics and other **qdiskd** parameters, refer to the qdisk(5) man page. If you require assistance understanding and using **qdiskd** for your site, contact an authorized Red Hat support representative.

If you need to use **qdiskd**, you should take into account the following considerations:

Cluster node votes

Each cluster node should have the same number of votes.

CMAN membership timeout value

The CMAN membership timeout value (the time a node needs to be unresponsive before CMAN considers that node to be dead, and not a member) should be at least two times that of the **qdiskd** membership timeout value. The reason is because the quorum daemon must detect failed nodes on its own, and can take much longer to do so than CMAN. The default value for CMAN membership timeout is 10 seconds. Other site-specific conditions may affect the relationship between the membership timeout values of CMAN and **qdiskd**. For assistance with adjusting the CMAN membership timeout value, contact an authorized Red Hat support representative.

Fencing

To ensure reliable fencing when using **qdiskd**, use power fencing. While other types of fencing (such as watchdog timers and software-based solutions to reboot a node internally) can be reliable for clusters not configured with **qdiskd**, they are not reliable for a cluster configured with **qdiskd**.

Maximum nodes

A cluster configured with **qdiskd** supports a maximum of 16 nodes. The reason for the limit is because of scalability; increasing the node count increases the amount of synchronous I/O contention on the shared quorum disk device.

Quorum disk device

A quorum disk device should be a shared block device with concurrent read/write access by all nodes in a cluster. The minimum size of the block device is 10 Megabytes. Examples of shared block devices that can be used by **qdiskd** are a multi-port SCSI RAID array, a Fibre Channel RAID SAN, or a RAID-configured iSCSI target. You can create a quorum disk device with **mkqdisk**, the Cluster Quorum Disk Utility. For information about using the utility refer to the mkqdisk(8) man page.

Note

Using JBOD as a quorum disk is not recommended. A JBOD cannot provide dependable performance and therefore may not allow a node to write to it quickly enough. If a node is unable to write to a quorum disk device quickly enough, the node is falsely evicted from a cluster.

2.8. Red Hat Cluster Suite and SELinux

Red Hat Cluster Suite supports SELinux states according to the Red Hat Enterprise Linux release level deployed in your cluster as follows:

- Red Hat Enterprise Linux 5.4 and earlier — **disabled** state only.
- Red Hat Enterprise Linux 5.5 and later — **enforcing** or **permissive** state with the SELinux policy type set to **targeted** (*or* with the **state** set to **disabled**).

Note

When using SELinux with Red Hat Cluster Suite in a VM environment, you should ensure that the SELinux boolean **fenced_can_network_connect** is persistently set to **on**. This allows the **fence_xvm** fencing agent to work properly, enabling the system to fence virtual machines.

For more information about SELinux, refer to *Deployment Guide* for Red Hat Enterprise Linux 5.

2.9. Multicast Addresses

Red Hat Cluster nodes communicate among each other using multicast addresses. Therefore, each network switch and associated networking equipment in a Red Hat Cluster must be configured to enable multicast addresses and support IGMP (Internet Group Management Protocol). Ensure that each network switch and associated networking equipment in a Red Hat Cluster are capable of supporting multicast addresses and IGMP; if they are, ensure that multicast addressing and IGMP are enabled. Without multicast and IGMP, not all nodes can participate in a cluster, causing the cluster to fail.

Note

Procedures for configuring network switches and associated networking equipment vary according each product. Refer to the appropriate vendor documentation or other information about configuring network switches and associated networking equipment to enable multicast addresses and IGMP.

Note

IPV6 is not supported for Cluster Suite in Red Hat Enterprise Linux 5.

2.10. Configuring the iptables Firewall to Allow Cluster Components

You can use the following filtering to allow multicast traffic through the **iptables** firewall for the various cluster components.

For **openais**, use the following filtering. Port 5405 is used to receive multicast traffic.

```
iptables -I INPUT -p udp -m state --state NEW -m multiport --dports 5404,5405 -j
ACCEPT
```

For **ricci**:

```
iptables -I INPUT -p tcp -m state --state NEW -m multiport --dports 11111 -j ACCEPT
```

For **modcluster**:

```
iptables -I INPUT -p tcp -m state --state NEW -m multiport --dports 16851 -j ACCEPT
```

For **gnbd**:

```
iptables -I INPUT -p tcp -m state --state NEW -m multiport --dports 14567 -j ACCEPT
```

For **luci**:

```
iptables -I INPUT -p tcp -m state --state NEW -m multiport --dports 8084 -j ACCEPT
```

For **DLM**:

```
iptables -I INPUT -p tcp -m state --state NEW -m multiport --dports 21064 -j ACCEPT
```

For **ccsd**:

```
iptables -I INPUT -p udp -m state --state NEW -m multiport --dports 50007 -j ACCEPT
iptables -I INPUT -p tcp -m state --state NEW -m multiport --dports 50008 -j ACCEPT
```

After executing these commands, run the following command.

```
service iptables save ; service iptables restart
```

In Red Hat Enterprise Linux 5, **rgmanager** does not access the network directly; **rgmanager** communication happens by means of **openais** network transport. Enabling **openais** allows **rgmanager** (or any **openais** clients) to work automatically.

2.11. Considerations for Using Conga

When using **Conga** to configure and manage your Red Hat Cluster, make sure that each computer running **luci** (the **Conga** user interface server) is running on the same network that the cluster is using for cluster communication. Otherwise, **luci** cannot configure the nodes to communicate on the right network. If the computer running **luci** is on another network (for example, a public network rather than a private network that the cluster is communicating on), contact an authorized Red Hat support representative to make sure that the appropriate host name is configured for each cluster node.

2.12. Configuring Virtual Machines in a Clustered Environment

When you configure your cluster with virtual machine resources, you should use the **rgmanager** tools to start and stop the virtual machines. Using **xm** or **virsh** to start the machine can result in the virtual machine running in more than one place, which can cause data corruption in the virtual machine.

To reduce the chances of administrators accidentally "double-starting" virtual machines by using both cluster and non-cluster tools in a clustered environment, you can configure your system as follows:

- Ensure that you are using the **rgmanager 2.0.52-1.el5_4.3** or later package release.
- Store the virtual machine configuration files in a non-default location.

Storing the virtual machine configuration files somewhere other than their default location makes it more difficult to accidentally start a virtual machine using **xm** or **virsh**, as the configuration file will be unknown out of the box to **libvirt** or the **xm** tool.

The non-default location for virtual machine configuration files may be anywhere. The advantage of using an NFS share or a shared GFS or GFS2 file system is that the administrator does not need to keep the configuration files in sync across the cluster members. However, it is also permissible to use a local directory as long as the administrator keeps the contents synchronized somehow cluster-wide.

In the cluster configuration, virtual machines may reference this non-default location by using the **path** attribute of a virtual machine resource. Note that the **path** attribute is a directory or set of directories separated by the colon ':' character, not a path to a specific file.

For more information on the attributes of a virtual machine resources, refer to Table C.23, “Virtual Machine”.

Chapter 3. Configuring Red Hat Cluster With Conga

This chapter describes how to configure Red Hat Cluster software using **Conga**, and consists of the following sections:

- Section 3.1, “Configuration Tasks”
- Section 3.2, “Starting **luci** and **ricci**”
- Section 3.3, “Creating A Cluster”
- Section 3.4, “Global Cluster Properties”
- Section 3.5, “Configuring Fence Devices”
- Section 3.6, “Configuring Cluster Members”
- Section 3.7, “Configuring a Failover Domain”
- Section 3.8, “Adding Cluster Resources”
- Section 3.9, “Adding a Cluster Service to the Cluster”
- Section 3.10, “Configuring Cluster Storage”

3.1. Configuration Tasks

Configuring Red Hat Cluster software with **Conga** consists of the following steps:

1. Configuring and running the **Conga** configuration user interface — the **luci** server. Refer to Section 3.2, “Starting **luci** and **ricci**”.
2. Creating a cluster. Refer to Section 3.3, “Creating A Cluster”.
3. Configuring global cluster properties. Refer to Section 3.4, “Global Cluster Properties”.
4. Configuring fence devices. Refer to Section 3.5, “Configuring Fence Devices”.
5. Configuring cluster members. Refer to Section 3.6, “Configuring Cluster Members”.
6. Creating failover domains. Refer to Section 3.7, “Configuring a Failover Domain”.
7. Creating resources. Refer to Section 3.8, “Adding Cluster Resources”.
8. Creating cluster services. Refer to Section 3.9, “Adding a Cluster Service to the Cluster”.
9. Configuring storage. Refer to Section 3.10, “Configuring Cluster Storage”.

3.2. Starting luci and ricci

To administer Red Hat Clusters with **Conga**, install and run **luci** and **ricci** as follows:

1. At each node to be administered by **Conga**, install the **ricci** agent. For example:

```
# yum install ricci
```

2. At each node to be administered by **Conga**, start **ricci**. For example:

```
# service ricci start
Starting ricci:                                            [  OK  ]
```

3. Select a computer to host **luci** and install the **luci** software on that computer. For example:

```
# yum install luci
```

Note

Typically, a computer in a server cage or a data center hosts **luci**; however, a cluster computer can host **luci**.

4. At the computer running **luci**, initialize the **luci** server using the **luci_admin init** command. For example:

```
# luci_admin init
Initializing the Luci server


Creating the 'admin' user

Enter password:  <Type password and press ENTER.>
Confirm password: <Re-type password and press ENTER.>

Please wait...
The admin password has been successfully set.
Generating SSL certificates...
Luci server has been successfully initialized


Restart the Luci server for changes to take effect
eg. service luci restart
```

5. Start **luci** using **service luci restart**. For example:

```
# service luci restart
Shutting down luci:                                        [  OK  ]
Starting luci: generating https SSL certificates...  done
                                                           [  OK  ]

Please, point your web browser to https://nano-01:8084 to access luci
```

6. At a Web browser, place the URL of the **luci** server into the URL address box and click **Go** (or the equivalent). The URL syntax for the **luci** server is **https://*luci_server_hostname*:8084**. The first time you access **luci**, two SSL certificate dialog boxes are displayed. Upon acknowledging the dialog boxes, your Web browser displays the **luci** login page.

3.3. Creating A Cluster

Creating a cluster with **luci** consists of selecting cluster nodes, entering their passwords, and submitting the request to create a cluster. If the node information and passwords are correct, **Conga** automatically installs software into the cluster nodes and starts the cluster. Create a cluster as follows:

1. As administrator of **luci**, select the **cluster** tab.
2. Click **Create a New Cluster**.
3. At the **Cluster Name** text box, enter a cluster name. The cluster name cannot exceed 15 characters. Add the node name and password for each cluster node. Enter the node name for each node in the **Node Hostname** column; enter the root password for each node in the **Root Password** column. Check the **Enable Shared Storage Support** checkbox if clustered storage is required.
4. Click **Submit**. Clicking **Submit** causes the following actions:
    a. Cluster software packages to be downloaded onto each cluster node.
    b. Cluster software to be installed onto each cluster node.
    c. Cluster configuration file to be created and propagated to each node in the cluster.
    d. Starting the cluster.

    A progress page shows the progress of those actions for each node in the cluster.

    When the process of creating a new cluster is complete, a page is displayed providing a configuration interface for the newly created cluster.

3.4. Global Cluster Properties

When a cluster is created, or if you select a cluster to configure, a cluster-specific page is displayed. The page provides an interface for configuring cluster-wide properties and detailed properties. You can configure cluster-wide properties with the tabbed interface below the cluster name. The interface provides the following tabs: **General**, **Fence**, **Multicast**, and **Quorum Partition**. To configure the parameters in those tabs, follow the steps in this section. If you do not need to configure parameters in a tab, skip the step for that tab.

1. **General** tab — This tab displays cluster name and provides an interface for configuring the configuration version and advanced cluster properties. The parameters are summarized as follows:
    - The **Cluster Name** text box displays the cluster name; it does not accept a cluster name change. You cannot change the cluster name. The only way to change the name of a Red Hat cluster is to create a new cluster configuration with the new name.
    - The **Configuration Version** value is set to **1** by default and is automatically incremented each time you modify your cluster configuration. However, if you need to set it to another value, you can specify it at the **Configuration Version** text box.
    - You can enter advanced cluster properties by clicking **Show advanced cluster properties**. Clicking **Show advanced cluster properties** reveals a list of advanced properties. You can click any advanced property for online help about the property.

    Enter the values required and click **Apply** for changes to take effect.
2. **Fence** tab — This tab provides an interface for configuring these **Fence Daemon Properties** parameters: **Post-Fail Delay** and **Post-Join Delay**. The parameters are summarized as follows:

- The **Post-Fail Delay** parameter is the number of seconds the fence daemon (**fenced**) waits before fencing a node (a member of the fence domain) after the node has failed. The **Post-Fail Delay** default value is **0**. Its value may be varied to suit cluster and network performance.
- The **Post-Join Delay** parameter is the number of seconds the fence daemon (**fenced**) waits before fencing a node after the node joins the fence domain. The **Post-Join Delay** default value is **3**. A typical setting for **Post-Join Delay** is between 20 and 30 seconds, but can vary according to cluster and network performance.

Enter values required and Click **Apply** for changes to take effect.

Note

For more information about **Post-Join Delay** and **Post-Fail Delay**, refer to the fenced(8) man page.

3. **Multicast** tab — This tab provides an interface for configuring these **Multicast Configuration** parameters: **Let cluster choose the multicast address** and **Specify the multicast address manually**. The default setting is **Let cluster choose the multicast address**. If you need to use a specific multicast address, click **Specify the multicast address manually**, enter a multicast address into the text box, and click **Apply** for changes to take effect.

Note

IPV6 is not supported for Cluster Suite in Red Hat Enterprise Linux 5.

If you do not specify a multicast address, the Red Hat Cluster software (specifically, **cman**, the Cluster Manager) creates one. It forms the upper 16 bits of the multicast address with 239.192 and forms the lower 16 bits based on the cluster ID.

Note

The cluster ID is a unique identifier that **cman** generates for each cluster. To view the cluster ID, run the **cman_tool status** command on a cluster node.

If you do specify a multicast address, you should use the 239.192.x.x series that **cman** uses. Otherwise, using a multicast address outside that range may cause unpredictable results. For example, using 224.0.0.x (which is "All hosts on the network") may not be routed correctly, or even routed at all by some hardware.

Note

If you specify a multicast address, make sure that you check the configuration of routers that cluster packets pass through. Some routers may take a long time to learn addresses, seriously impacting cluster performance.

4. **Quorum Partition** tab — This tab provides an interface for configuring these **Quorum Partition Configuration** parameters: **Do not use a Quorum Partition**, **Use a Quorum Partition**, **Interval**, **Votes**, **TKO**, **Minimum Score**, **Device**, **Label**, and **Heuristics**. The **Do not use a Quorum Partition** parameter is enabled by default. Table 3.1, "Quorum-Disk Parameters" describes the parameters. If you need to use a quorum disk, click **Use a Quorum Partition**, enter quorum disk parameters, click **Apply**, and restart the cluster for the changes to take effect.

Important

Quorum-disk parameters and heuristics depend on the site environment and the special requirements needed. To understand the use of quorum-disk parameters and heuristics, refer to the qdisk(5) man page. If you require assistance understanding and using quorum disk, contact an authorized Red Hat support representative.

Note

Clicking **Apply** on the **Quorum Partition** tab propagates changes to the cluster configuration file (**/etc/cluster/cluster.conf**) in each cluster node. However, for the quorum disk to operate, you must restart the cluster (refer to Section 4.1, "Starting, Stopping, and Deleting Clusters").

Table 3.1. Quorum-Disk Parameters

Parameter	Description
Do not use a Quorum Partition	Disables quorum partition. Disables quorum-disk parameters in the **Quorum Partition** tab.
Use a Quorum Partition	Enables quorum partition. Enables quorum-disk parameters in the **Quorum Partition** tab.
Interval	The frequency of read/write cycles, in seconds.
Votes	The number of votes the quorum daemon advertises to CMAN when it has a high enough score.
TKO	The number of cycles a node must miss to be declared dead.
Minimum Score	The minimum score for a node to be considered "alive". If omitted or set to 0, the default function, **floor((*n*+1)/2)**, is used, where *n* is the sum of the heuristics scores. The **Minimum Score** value must never exceed the sum of the heuristic scores; otherwise, the quorum disk cannot be available.
Device	The storage device the quorum daemon uses. The device must be the same on all nodes.
Label	Specifies the quorum disk label created by the **mkqdisk** utility. If this field contains an entry, the label overrides the **Device** field. If this field is used, the quorum daemon reads **/proc/partitions** and checks for qdisk signatures on every block device found, comparing the label against the specified label. This is useful in configurations where the quorum device name differs among nodes.

Parameter	Description
Heuristics	**Path to Program** — The program used to determine if this heuristic is alive. This can be anything that can be executed by **/bin/sh -c**. A return value of 0 indicates success; anything else indicates failure. This field is required. **Interval** — The frequency (in seconds) at which the heuristic is polled. The default interval for every heuristic is 2 seconds. **Score** — The weight of this heuristic. Be careful when determining scores for heuristics. The default score for each heuristic is 1.
Apply	Propagates the changes to the cluster configuration file (**/etc/cluster/cluster.conf**) in each cluster node.

3.5. Configuring Fence Devices

Configuring fence devices consists of creating, modifying, and deleting fence devices. Creating a fence device consists of selecting a fence device type and entering parameters for that fence device (for example, name, IP address, login, and password). Modifying a fence device consists of selecting an existing fence device and changing parameters for that fence device. Deleting a fence device consists of selecting an existing fence device and deleting it.

Note

If you are creating a new cluster, you can create fence devices when you configure cluster nodes. Refer to Section 3.6, “Configuring Cluster Members”.

With **Conga** you can create shared and non-shared fence devices. For information on supported fence devices and their parameters, refer to Appendix B, *Fence Device Parameters*.

This section provides procedures for the following tasks:

- Creating *shared* fence devices — Refer to Section 3.5.1, “Creating a Shared Fence Device”. The procedures apply *only* to creating shared fence devices. You can create *non-shared* (and shared) fence devices while configuring nodes (refer to Section 3.6, “Configuring Cluster Members”).
- Modifying or deleting fence devices — Refer to Section 3.5.2, “Modifying or Deleting a Fence Device”. The procedures apply to both shared and non-shared fence devices.

The starting point of each procedure is at the cluster-specific page that you navigate to from **Choose a cluster to administer** displayed on the **cluster** tab.

3.5.1. Creating a Shared Fence Device

To create a shared fence device, follow these steps:

1. At the detailed menu for the cluster (below the **clusters** menu), click **Shared Fence Devices**. Clicking **Shared Fence Devices** causes the display of the fence devices for a cluster and causes the display of menu items for fence device configuration: **Add a Fence Device** and **Configure a Fence Device**.

Note

If this is an initial cluster configuration, no fence devices have been created, and therefore none are displayed.

2. Click **Add a Fence Device**. Clicking **Add a Fence Device** causes the **Add a Sharable Fence Device** page to be displayed (refer to Figure 3.1, "Fence Device Configuration").

Figure 3.1. Fence Device Configuration

3. At the **Add a Sharable Fence Device** page, click the drop-down box under **Fencing Type** and select the type of fence device to configure.

4. Specify the information in the **Fencing Type** dialog box according to the type of fence device. Refer to Appendix B, *Fence Device Parameters* for more information about fence device parameters.

5. Click **Add this shared fence device**.

   Clicking **Add this shared fence device** causes a progress page to be displayed temporarily. After the fence device has been added, the detailed cluster properties menu is updated with the fence device under **Configure a Fence Device**.

3.5.2. Modifying or Deleting a Fence Device

To modify or delete a fence device, follow these steps:

1. At the detailed menu for the cluster (below the **clusters** menu), click **Shared Fence Devices**. Clicking **Shared Fence Devices** causes the display of the fence devices for a cluster and causes the display of menu items for fence device configuration: **Add a Fence Device** and **Configure a Fence Device**.

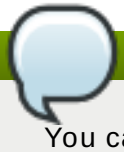
2. Click **Configure a Fence Device**. Clicking **Configure a Fence Device** causes the display of a list of fence devices under **Configure a Fence Device**.

3. Click a fence device in the list. Clicking a fence device in the list causes the display of a **Fence Device Form** page for the fence device selected from the list.

4. Either modify or delete the fence device as follows:

   - To modify the fence device, enter changes to the parameters displayed. Refer to Appendix B, *Fence Device Parameters* for more information about fence device parameters. Click **Update this fence device** and wait for the configuration to be updated.

   - To delete the fence device, click **Delete this fence device** and wait for the configuration to be updated.



Note

You can create shared fence devices on the node configuration page, also. However, you can only modify or delete a shared fence device via **Shared Fence Devices** at the detailed menu for the cluster (below the **clusters** menu).

3.6. Configuring Cluster Members

Configuring cluster members consists of initially configuring nodes in a newly configured cluster, adding members, and deleting members. The following sections provide procedures for initial configuration of nodes, adding nodes, and deleting nodes:

- Section 3.6.1, “Initially Configuring Members”
- Section 3.6.2, “Adding a Member to a Running Cluster”
- Section 3.6.3, “Deleting a Member from a Cluster”

3.6.1. Initially Configuring Members

Creating a cluster consists of selecting a set of nodes (or members) to be part of the cluster. Once you have completed the initial step of creating a cluster and creating fence devices, you need to configure cluster nodes. To initially configure cluster nodes after creating a new cluster, follow the steps in this section. The starting point of the procedure is at the cluster-specific page that you navigate to from **Choose a cluster to administer** displayed on the **cluster** tab.

1. At the detailed menu for the cluster (below the **clusters** menu), click **Nodes**. Clicking **Nodes** causes the display of an **Add a Node** element and a **Configure** element with a list of the nodes already configured in the cluster.
2. Click a link for a node at either the list in the center of the page or in the list in the detailed menu under the **clusters** menu. Clicking a link for a node causes a page to be displayed for that link showing how that node is configured.
3. At the bottom of the page, under **Main Fencing Method**, click **Add a fence device to this level**.
4. Select a fence device and provide parameters for the fence device (for example port number).

Note

You can choose from an existing fence device or create a new fence device.

5. Click **Update main fence properties** and wait for the change to take effect.

3.6.2. Adding a Member to a Running Cluster

To add a member to a running cluster, follow the steps in this section. The starting point of the procedure is at the cluster-specific page that you navigate to from **Choose a cluster to administer** displayed on the **cluster** tab.

1. At the detailed menu for the cluster (below the **clusters** menu), click **Nodes**. Clicking **Nodes** causes the display of an **Add a Node** element and a **Configure** element with a list of the nodes already configured in the cluster. (In addition, a list of the cluster nodes is displayed in the center of the page.)
2. Click **Add a Node**. Clicking **Add a Node** causes the display of the **Add a node to *cluster name*** page.
3. At that page, enter the node name in the **Node Hostname** text box; enter the root password in the **Root Password** text box. Check the **Enable Shared Storage Support** checkbox if clustered storage is required. If you want to add more nodes, click **Add another entry** and enter node name and password for the each additional node.
4. Click **Submit**. Clicking **Submit** causes the following actions:
    a. Cluster software packages to be downloaded onto the added node.
    b. Cluster software to be installed (or verification that the appropriate software packages are installed) onto the added node.
    c. Cluster configuration file to be updated and propagated to each node in the cluster — including the added node.
    d. Joining the added node to cluster.

A progress page shows the progress of those actions for each added node.

5. When the process of adding a node is complete, a page is displayed providing a configuration interface for the cluster.
6. At the detailed menu for the cluster (below the **clusters** menu), click **Nodes**. Clicking **Nodes** causes the following displays:
   - A list of cluster nodes in the center of the page
   - The **Add a Node** element and the **Configure** element with a list of the nodes configured in the cluster at the detailed menu for the cluster (below the **clusters** menu)
7. Click the link for an added node at either the list in the center of the page or in the list in the detailed menu under the **clusters** menu. Clicking the link for the added node causes a page to be displayed for that link showing how that node is configured.
8. At the bottom of the page, under **Main Fencing Method**, click **Add a fence device to this level**.
9. Select a fence device and provide parameters for the fence device (for example port number).

Note

You can choose from an existing fence device or create a new fence device.

10. Click **Update main fence properties** and wait for the change to take effect.

3.6.3. Deleting a Member from a Cluster

To delete a member from an existing cluster that is currently in operation, follow the steps in this section. The starting point of the procedure is at the **Choose a cluster to administer** page (displayed on the **cluster** tab).

1. Click the link of the node to be deleted. Clicking the link of the node to be deleted causes a page to be displayed for that link showing how that node is configured.

Note

To allow services running on a node to fail over when the node is deleted, skip the next step.

2. Disable or relocate each service that is running on the node to be deleted:

Note

Repeat this step for each service that needs to be disabled or started on another node.

   a. Under **Services on this Node**, click the link for a service. Clicking that link cause a configuration page for that service to be displayed.

b. On that page, at the **Choose a task**drop-down box, choose to either disable the service are start it on another node and click **Go**.

c. Upon confirmation that the service has been disabled or started on another node, click the **cluster** tab. Clicking the **cluster** tab causes the **Choose a cluster to administer** page to be displayed.

d. At the **Choose a cluster to administer** page, click the link of the node to be deleted. Clicking the link of the node to be deleted causes a page to be displayed for that link showing how that node is configured.

3. On that page, at the **Choose a task**drop-down box, choose **Delete this node** and click **Go**. When the node is deleted, a page is displayed that lists the nodes in the cluster. Check the list to make sure that the node has been deleted.

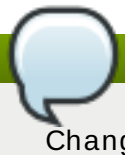
3.7. Configuring a Failover Domain

A failover domain is a named subset of cluster nodes that are eligible to run a cluster service in the event of a node failure. A failover domain can have the following characteristics:

- Unrestricted — Allows you to specify that a subset of members are preferred, but that a cluster service assigned to this domain can run on any available member.
- Restricted — Allows you to restrict the members that can run a particular cluster service. If none of the members in a restricted failover domain are available, the cluster service cannot be started (either manually or by the cluster software).
- Unordered — When a cluster service is assigned to an unordered failover domain, the member on which the cluster service runs is chosen from the available failover domain members with no priority ordering.
- Ordered — Allows you to specify a preference order among the members of a failover domain. The member at the top of the list is the most preferred, followed by the second member in the list, and so on.
- Failback — Allows you to specify whether a service in the failover domain should fail back to the node that it was originally running on before that node failed. Configuring this characteristic is useful in circumstances where a node repeatedly fails and is part of an ordered failover domain. In that circumstance, if a node is the preferred node in a failover domain, it is possible for a service to fail over and fail back repeatedly between the preferred node and another node, causing severe impact on performance.

Note

The failback characteristic is applicable only if ordered failover is configured.



Note

Changing a failover domain configuration has no effect on currently running services.

Note

Failover domains are *not* required for operation.

By default, failover domains are unrestricted and unordered.

In a cluster with several members, using a restricted failover domain can minimize the work to set up the cluster to run a cluster service (such as **httpd**), which requires you to set up the configuration identically on all members that run the cluster service). Instead of setting up the entire cluster to run the cluster service, you must set up only the members in the restricted failover domain that you associate with the cluster service.

Note

To configure a preferred member, you can create an unrestricted failover domain comprising only one cluster member. Doing that causes a cluster service to run on that cluster member primarily (the preferred member), but allows the cluster service to fail over to any of the other members.

The following sections describe adding a failover domain and modifying a failover domain:

- Section 3.7.1, “Adding a Failover Domain”
- Section 3.7.2, “Modifying a Failover Domain”

3.7.1. Adding a Failover Domain

To add a failover domain, follow the steps in this section. The starting point of the procedure is at the cluster-specific page that you navigate to from **Choose a cluster to administer** displayed on the **cluster** tab.

1. At the detailed menu for the cluster (below the **clusters** menu), click **Failover Domains**. Clicking **Failover Domains** causes the display of failover domains with related services and the display of menu items for failover domains: **Add a Failover Domain** and **Configure a Failover Domain** .
2. Click **Add a Failover Domain**. Clicking **Add a Failover Domain** causes the display of the **Add a Failover Domain** page.
3. At the **Add a Failover Domain** page, specify a failover domain name at the **Failover Domain Name** text box.

   Note

   The name should be descriptive enough to distinguish its purpose relative to other names used in your cluster.

4. To enable setting failover priority of the members in the failover domain, click the **Prioritized** checkbox. With **Prioritized** checked, you can set the priority value, **Priority**, for each node selected as members of the failover domain.

5. To restrict failover to members in this failover domain, click the checkbox next to **Restrict failover to this domain's members**. With **Restrict failover to this domain's members** checked, services assigned to this failover domain fail over only to nodes in this failover domain.

6. To specify that a node does not fail back in this failover domain, click the checkbox next to **Do not fail back services in this domain**. With **Do not fail back services in this domain** checked, if a service fails over from a preferred node, the service does not fail back to the original node once it has recovered.

7. Configure members for this failover domain. Under **Failover domain membership**, click the **Member** checkbox for each node that is to be a member of the failover domain. If **Prioritized** is checked, set the priority in the **Priority** text box for each member of the failover domain.

8. Click **Submit**. Clicking **Submit** causes a progress page to be displayed followed by the display of the **Failover Domain Form** page. That page displays the added resource and includes the failover domain in the cluster menu to the left under **Domain**.

9. To make additional changes to the failover domain, continue modifications at the **Failover Domain Form** page and click **Submit** when you are done.

3.7.2. Modifying a Failover Domain

To modify a failover domain, follow the steps in this section. The starting point of the procedure is at the cluster-specific page that you navigate to from **Choose a cluster to administer** displayed on the **cluster** tab.

1. At the detailed menu for the cluster (below the **clusters** menu), click **Failover Domains**. Clicking **Failover Domains** causes the display of failover domains with related services and the display of menu items for failover domains: **Add a Failover Domain** and **Configure a Failover Domain** .

2. Click **Configure a Failover Domain**. Clicking **Configure a Failover Domain** causes the display of failover domains under **Configure a Failover Domain** at the detailed menu for the cluster (below the **clusters** menu).

3. At the detailed menu for the cluster (below the **clusters** menu), click the failover domain to modify. Clicking the failover domain causes the display of the **Failover Domain Form** page. At the **Failover Domain Form** page, you can modify the failover domain name, prioritize failover, restrict failover to this domain, and modify failover domain membership.

4. Modifying failover name — To change the failover domain name, modify the text at the **Failover Domain Name** text box.

Note

The name should be descriptive enough to distinguish its purpose relative to other names used in your cluster.

5. Failover priority — To enable or disable prioritized failover in this failover domain, click the **Prioritized** checkbox. With **Prioritized** checked, you can set the priority value, **Priority**, for each node selected as members of the failover domain. With **Prioritized** *not* checked, setting priority levels is disabled for this failover domain.

6. Restricted failover — To enable or disable restricted failover for members in this failover domain, click the checkbox next to **Restrict failover to this domain's members**. With **Restrict failover to this domain's members** checked, services assigned to this failover domain fail over only to nodes in this failover domain. With **Restrict failover to this domain's members** *not* checked, services assigned to this failover domain can fail over to nodes outside this failover domain.

7. Failback — To enable or disable failback in a failover domain, click the checkbox next to **Do not fail back services in this domain**. With **Do not fail back services in this domain** checked, if a service fails over from a preferred node, the service does not fail back to the original node once it has recovered.

8. Modifying failover domain membership — Under **Failover domain membership**, click the **Member**checkbox for each node that is to be a member of the failover domain. A checked box for a node means that the node is a member of the failover domain. If **Prioritized** is checked, you can adjust the priority in the **Priority** text box for each member of the failover domain.

9. Click **Submit**. Clicking **Submit** causes a progress page to be displayed followed by the display of the **Failover Domain Form** page. That page displays the added resource and includes the failover domain in the cluster menu to the left under **Domain**.

10. To make additional changes to the failover domain, continue modifications at the **Failover Domain Form** page and click **Submit** when you are done.

3.8. Adding Cluster Resources

To add a cluster resource, follow the steps in this section. The starting point of the procedure is at the cluster-specific page that you navigate to from **Choose a cluster to administer** displayed on the **cluster** tab.

1. At the detailed menu for the cluster (below the **clusters** menu), click **Resources**. Clicking **Resources** causes the display of resources in the center of the page and causes the display of menu items for resource configuration: **Add a Resource** and **Configure a Resource**.

2. Click **Add a Resource**. Clicking **Add a Resource** causes the **Add a Resource** page to be displayed.

3. At the **Add a Resource** page, click the drop-down box under **Select a Resource Type** and select the type of resource to configure. Appendix C, *HA Resource Parameters* describes resource parameters.

4. Click **Submit**. Clicking **Submit** causes a progress page to be displayed followed by the display of **Resources for*cluster name*** page. That page displays the added resource (and other resources).

3.9. Adding a Cluster Service to the Cluster

To add a cluster service to the cluster, follow the steps in this section. The starting point of the procedure is at the cluster-specific page that you navigate to from **Choose a cluster to administer** displayed on the **cluster** tab.

1. At the detailed menu for the cluster (below the **clusters** menu), click **Services**. Clicking **Services** causes the display of services in the center of the page and causes the display of menu items for services configuration: **Add a Service**, **Add a Virtual Machine Service**, and **Configure a Service**.

2. To configure any service other than a virtual machine service, Click **Add a Service**. Clicking **Add a Service** causes the **Add a Service** page to be displayed.

3. On the **Add a Service** page, at the **Service name** text box, type the name of the service.

Note

Use a descriptive name that clearly distinguishes the service from other services in the cluster.

4. Below the **Service name** text box, enter the following parameters for this service.

   - **Automatically start this service** — When the checkbox is checked, the service is started automatically when a cluster is started and running. If the checkbox is *not* checked, the service must be started manually any time the cluster comes up from the stopped state.

   - **Enable NFS lock workarounds** — Setting this option will release NFS locks on a file system in a soft attempt to unmount a file system, which may be necessary if your filesystem is exported via NFS and occasionally fails to unmount (either during shutdown or service relocation). You can also enable NFS daemon and lock workarounds for individual file system resources, which will result in a hard attempt to unmount a file system, as described in the table of file system resource parameters, Table C.3, “File System”, and the table of GFS resource parameters, Table C.4, “GFS”.

   - **Run exclusive** — If enabled, this service (resource group) can only be relocated to run on another node exclusively; that is, to run on a node that has no other services running on it. If no nodes are available for a service to run exclusively, the service is not restarted after a failure. Additionally, other services do not automatically relocate to a node running this service as ***Run exclusive***. You can override this option by manual start or relocate operations.

   - **Failover Domain** — List of cluster members to try in the event that a service fails. For information on configuring a failover domain with Conga, refer to Section 3.7, “Configuring a Failover Domain”.

   - **Recovery policy** Provides the following options:

     Disable — Disables the resource group if any component fails.

     Relocate — Tries to restart service in another node; that is, it does not try to restart in the current node.

     Restart — Tries to restart failed parts of this service locally (in the current node) before trying to relocate (default) to service to another node.

     Restart-Disable — (Red Hat Enterprise Linux release 5.6 and later) The service will be restarted in place if it fails. However, if restarting the service fails the service will be disabled instead of being moved to another host in the cluster.

   - In addition, you can specify the **Maximum number of restart failures before relocating** and the **Length of time in seconds after which to forget a restart**.

5. Add a resource to the service; click **Add a resource to this service**. Clicking **Add a resource to this service** causes the display of two drop-down boxes: **Add a new local resource** and **Use an existing global resource**. Adding a new local resource adds a resource that is available *only* to this service. The process of adding a local resource is the same as adding a global resource described in Section 3.8, "Adding Cluster Resources". Adding a global resource adds a resource that has been previously added as a global resource (refer to Section 3.8, "Adding Cluster Resources").

6. At the drop-down box of either **Add a new local resource** or **Use an existing global resource**, select the resource to add and configure it according to the options presented. (The options are the same as described in Section 3.8, "Adding Cluster Resources".)

Note

If you are adding a Samba-service resource, connect a Samba-service resource directly to the service, *not* to a resource within a service.

7. If you want to add resources to that resource, click **Add a child**. Clicking **Add a child** causes the display of additional options to local and global resources. You can continue adding children resources to the resource to suit your requirements. To view children resources, click the triangle icon to the left of **Show Children**.

8. When you have completed adding resources to the service, and have completed adding children resources to resources, click **Submit**. Clicking **Submit** causes a progress page to be displayed followed by a page displaying the added service (and other services).

Note

To verify the existence of the IP service resource used in a cluster service, you must use the **/sbin/ip addr list** command on a cluster node. The following output shows the **/sbin/ip addr list** command executed on a node running a cluster service:

```
1: lo: <LOOPBACK,UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP> mtu 1356 qdisc pfifo_fast qlen 1000
    link/ether 00:05:5d:9a:d8:91 brd ff:ff:ff:ff:ff:ff
    inet 10.11.4.31/22 brd 10.11.7.255 scope global eth0
    inet6 fe80::205:5dff:fe9a:d891/64 scope link
    inet 10.11.4.240/22 scope global secondary eth0
       valid_lft forever preferred_lft forever
```

Configuring a Virtual Machine Service

To configure a virtual machine service, after clicking **Services** you can click **Add a Virtual Machine Service**. Enter the virtual machine resource parameters. For a description of the virtual machine parameters, refer to Table C.23, "Virtual Machine". When you have completed adding the virtual machine resource parameters, click **Create Virtual Machine Service**.

3.10. Configuring Cluster Storage

To configure storage for a cluster, click the **storage** tab. Clicking that tab causes the display of the **Welcome to Storage Configuration Interface** page.

The **storage** tab allows you to monitor and configure storage on remote systems. It provides a means for configuring disk partitions, logical volumes (clustered and single system use), file system parameters, and mount points. The **storage** tab provides an interface for setting up shared storage for clusters and offers GFS and other file systems as file system options. When a you select the **storage** tab, the **Welcome to Storage Configuration Interface** page shows a list of systems available to the you in a navigation table to the left. A small form allows you to choose a storage unit size to suit your preference. That choice is persisted and can be changed at any time by returning to this page. In addition, you can change the unit type on specific configuration forms throughout the storage user interface. This general choice allows you to avoid difficult decimal representations of storage size (for example, if you know that most of your storage is measured in gigabytes, terabytes, or other more familiar representations).

Additionally, the **Welcome to Storage Configuration Interface** page lists systems that you are authorized to access, but currently are unable to administer because of a problem. Examples of problems:

- A computer is unreachable via the network.
- A computer has been re-imaged and the **luci** server admin must re-authenticate with the **ricci** agent on the computer.

A reason for the trouble is displayed if the storage user interface can determine it.

Only those computers that the user is privileged to administer is shown in the main navigation table. If you have no permissions on any computers, a message is displayed.

After you select a computer to administer, a general properties page is displayed for the computer. This page is divided into three sections:

- **Hard Drives**
- **Partitions**
- **Volume Groups**

Each section is set up as an expandable tree, with links to property sheets for specific devices, partitions, and storage entities.

Configure the storage for your cluster to suit your cluster requirements. If you are configuring Red Hat GFS, configure clustered logical volumes first, using CLVM. For more information about CLVM and GFS refer to Red Hat documentation for those products.

Note

Shared storage for use in Red Hat Cluster Suite requires that you be running the cluster logical volume manager daemon (**clvmd**) or the High Availability Logical Volume Management agents (HA-LVM). If you are not able to use either the **clvmd** daemon or HA-LVM for operational reasons or because you do not have the correct entitlements, you must not use single-instance LVM on the shared disk as this may result in data corruption. If you have any concerns please contact your Red Hat service representative.

Chapter 4. Managing Red Hat Cluster With Conga

This chapter describes various administrative tasks for managing a Red Hat Cluster and consists of the following sections:

- Section 4.1, “Starting, Stopping, and Deleting Clusters”
- Section 4.2, “Managing Cluster Nodes”
- Section 4.3, “Managing High-Availability Services”
- Section 4.4, “Backing Up and Restoring the luci Configuration”
- Section 4.5, “Diagnosing and Correcting Problems in a Cluster”

4.1. Starting, Stopping, and Deleting Clusters

You can perform the following cluster-management functions through the **luci** server component of **Conga**:

- Restart a cluster.
- Start a cluster.
- Stop a cluster.
- Delete a cluster.

To perform one of the functions in the preceding list, follow the steps in this section. The starting point of the procedure is at the **cluster** tab (at the **Choose a cluster to administer** page).

1. At the right of the **Cluster Name** for each cluster listed on the **Choose a cluster to administer** page is a drop-down box. By default, the drop-down box is set to **Restart this cluster**. Clicking the drop-down box box reveals all the selections available: **Restart this cluster**, **Stop this cluster/Start this cluster**, and **Delete this cluster**. The actions of each function are summarized as follows:
   - **Restart this cluster** — Selecting this action causes the cluster to be restarted. You can select this action for any state the cluster is in.
   - **Stop this cluster/Start this cluster** — **Stop this cluster** is available when a cluster is running. **Start this cluster** is available when a cluster is stopped.

     Selecting **Stop this cluster** shuts down cluster software in all cluster nodes.

     Selecting **Start this cluster** starts cluster software.
   - **Delete this cluster** — Selecting this action halts a running cluster, disables cluster software from starting automatically, and removes the cluster configuration file from each node. You can select this action for any state the cluster is in. Deleting a cluster frees each node in the cluster for use in another cluster.
2. Select one of the functions and click **Go**.
3. Clicking **Go** causes a progress page to be displayed. When the action is complete, a page is displayed showing either of the following pages according to the action selected:

- For **Restart this cluster** and **Stop this cluster/Start this cluster** — Displays a page with the list of nodes for the cluster.
- For **Delete this cluster** — Displays the **Choose a cluster to administer** page in the **cluster** tab, showing a list of clusters.

4.2. Managing Cluster Nodes

You can perform the following node-management functions through the **luci** server component of **Conga**:

- Make a node leave or join a cluster.
- Fence a node.
- Reboot a node.
- Delete a node.

To perform one the functions in the preceding list, follow the steps in this section. The starting point of the procedure is at the cluster-specific page that you navigate to from **Choose a cluster to administer** displayed on the **cluster** tab.

1. At the detailed menu for the cluster (below the **clusters** menu), click **Nodes**. Clicking **Nodes** causes the display of nodes in the center of the page and causes the display of an **Add a Node** element and a **Configure** element with a list of the nodes already configured in the cluster.
2. At the right of each node listed on the page displayed from the preceding step, click the **Choose a task** drop-down box. Clicking **Choose a task** drop-down box reveals the following selections: **Have node leave cluster/Have node join cluster**, **Fence this node**, **Reboot this node**, and **Delete**. The actions of each function are summarized as follows:
   - **Have node leave cluster/Have node join cluster** — **Have node leave cluster** is available when a node has joined of a cluster. **Have node join cluster** is available when a node has left a cluster.

     Selecting **Have node leave cluster** shuts down cluster software and makes the node leave the cluster. Making a node leave a cluster prevents the node from automatically joining the cluster when it is rebooted.

     Selecting **Have node join cluster** starts cluster software and makes the node join the cluster. Making a node join a cluster allows the node to automatically join the cluster when it is rebooted.
   - **Fence this node** — Selecting this action causes the node to be fenced according to how the node is configured to be fenced.
   - **Reboot this node** — Selecting this action causes the node to be rebooted.
   - **Delete** — Selecting this action causes the node to be deleted from the cluster configuration. It also stops all cluster services on the node, and deletes the **cluster.conf** file from **/etc/cluster/**.
3. Select one of the functions and click **Go**.

4. Clicking **Go** causes a progress page to be displayed. When the action is complete, a page is displayed showing the list of nodes for the cluster.

4.3. Managing High-Availability Services

You can perform the following management functions for high-availability services through the **luci** server component of **Conga**:

- Configure a service.
- Stop or start a service.
- Restart a service.
- Delete a service

To perform one the functions in the preceding list, follow the steps in this section. The starting point of the procedure is at the cluster-specific page that you navigate to from **Choose a cluster to administer** displayed on the **cluster** tab.

1. At the detailed menu for the cluster (below the **clusters** menu), click **Services**. Clicking **Services** causes the display of services for the cluster in the center of the page.
2. At the right of each service listed on the page, click the **Choose a task** drop-down box. Clicking **Choose a task** drop-down box reveals the following selections depending on if the service is running:
   - If service is running — **Configure this service**, **Restart this service**, and **Stop this service**.
   - If service is not running — **Configure this service**, **Start this service**, and **Delete this service**.

   The actions of each function are summarized as follows:
   - **Configure this service** — **Configure this service** is available when the service is running or not running. Selecting **Configure this service** causes the services configuration page for the service to be displayed. On that page, you can change the configuration of the service. For example, you can add a resource to the service. (For more information about adding resources and services, refer to Section 3.8, “Adding Cluster Resources” and Section 3.9, “Adding a Cluster Service to the Cluster”.) In addition, a drop-down box on the page provides other functions depending on if the service is running.

     When a service is running, the drop-down box provides the following functions: restarting, disabling, and relocating the service.

     When a service is not running, the drop-down box on the configuration page provides the following functions: enabling and deleting the service.

     If you are making configuration changes, save the changes by clicking **Save**. Clicking **Save** causes a progress page to be displayed. When the change is complete, another page is displayed showing a list of services for the cluster.

     If you have selected one of the functions in the drop-down box on the configuration page, click **Go**. Clicking **Go** causes a progress page to be displayed. When the change is complete, another page is displayed showing a list of services for the cluster.

- **Restart this service** and **Stop this service** — These selections are available when the service is running. Select either function and click **Go** to make the change take effect. Clicking **Go** causes a progress page to be displayed. When the change is complete, another page is displayed showing a list of services for the cluster.
- **Start this service** and **Delete this service** — These selections are available when the service is not running. Select either function and click **Go** to make the change take effect. Clicking **Go** causes a progress page to be displayed. When the change is complete, another page is displayed showing a list of services for the cluster.

4.4. Backing Up and Restoring the luci Configuration

You can use the following procedure to make a backup of the **luci** database, which is stored in the **/var/lib/luci/var/Data.fs** file. This is not the cluster configuration itself, which is stored in the **cluster.conf** file. Instead, it contains the list of users and clusters and related properties that **luci** maintains.

1. Execute **service luci stop**.
2. Execute **luci_admin backup [*backup_xml_file_path*]**.

   Specifying the *backup_xml_file_path* is optional. If you do not specify a file path, the backup file will be written to **/var/lib/luci/var/luci_backup.xml**.
3. Execute **service luci start**.

Use the following procedure to restore a **luci** database.

1. Execute **service luci stop**.
2. Execute **luci_admin restore -r *backup_xml_file_path***.

   If you do not specify a backup path argument, the command uses **/var/lib/luci/var/luci_backup.xml**.

   Specifying the **-r** option indicates that you will replace all configuration with the configuration specified in the backup file. If you do not specify this option, the default behavior (which you can also specify with the **-u**) is to merge any additional configuration information from the XML backup into the current database.
3. Execute **service luci start**.

4.5. Diagnosing and Correcting Problems in a Cluster

For information about diagnosing and correcting problems in a cluster, contact an authorized Red Hat support representative.

Chapter 5. Configuring Red Hat Cluster With system-config-cluster

This chapter describes how to configure Red Hat Cluster software using **system-config-cluster**, and consists of the following sections:

- Section 5.1, “Configuration Tasks”
- Section 5.2, “Starting the **Cluster Configuration Tool**”
- Section 5.3, “Configuring Cluster Properties”
- Section 5.4, “Configuring Fence Devices”
- Section 5.5, “Adding and Deleting Members”
- Section 5.6, “Configuring a Failover Domain”
- Section 5.7, “Adding Cluster Resources”
- Section 5.8, “Adding a Cluster Service to the Cluster”
- Section 5.9, “Propagating The Configuration File: New Cluster”
- Section 5.10, “Starting the Cluster Software”

Note

While **system-config-cluster** provides several convenient tools for configuring and managing a Red Hat Cluster, the newer, more comprehensive tool, **Conga**, provides more convenience and flexibility than **system-config-cluster**. You may want to consider using **Conga** instead (refer to Chapter 3, *Configuring Red Hat Cluster With **Conga***** and Chapter 4, *Managing Red Hat Cluster With **Conga***).

5.1. Configuration Tasks

Configuring Red Hat Cluster software with **system-config-cluster** consists of the following steps:

1. Starting the **Cluster Configuration Tool**, **system-config-cluster**. Refer to Section 5.2, “Starting the **Cluster Configuration Tool**”.
2. Configuring cluster properties. Refer to Section 5.3, “Configuring Cluster Properties”.
3. Creating fence devices. Refer to Section 5.4, “Configuring Fence Devices”.
4. Creating cluster members. Refer to Section 5.5, “Adding and Deleting Members”.
5. Creating failover domains. Refer to Section 5.6, “Configuring a Failover Domain”.
6. Creating resources. Refer to Section 5.7, “Adding Cluster Resources”.
7. Creating cluster services.

   Refer to Section 5.8, “Adding a Cluster Service to the Cluster”.

8. Propagating the configuration file to the other nodes in the cluster.

   Refer to Section 5.9, "Propagating The Configuration File: New Cluster".

9. Starting the cluster software. Refer to Section 5.10, "Starting the Cluster Software".

5.2. Starting the Cluster Configuration Tool

You can start the **Cluster Configuration Tool** by logging in to a cluster node as root with the **ssh -Y** command and issuing the **system-config-cluster** command. For example, to start the **Cluster Configuration Tool** on cluster node nano-01, do the following:

1. Log in to a cluster node and run **system-config-cluster**. For example:

```
$  ssh -Y root@nano-01
  .
  .
  .
# system-config-cluster
```

2. If this is the first time you have started the **Cluster Configuration Tool**, the program prompts you to either open an existing configuration or create a new one. Click **Create New Configuration** to start a new configuration file (refer to Figure 5.1, "Starting a New Configuration File").

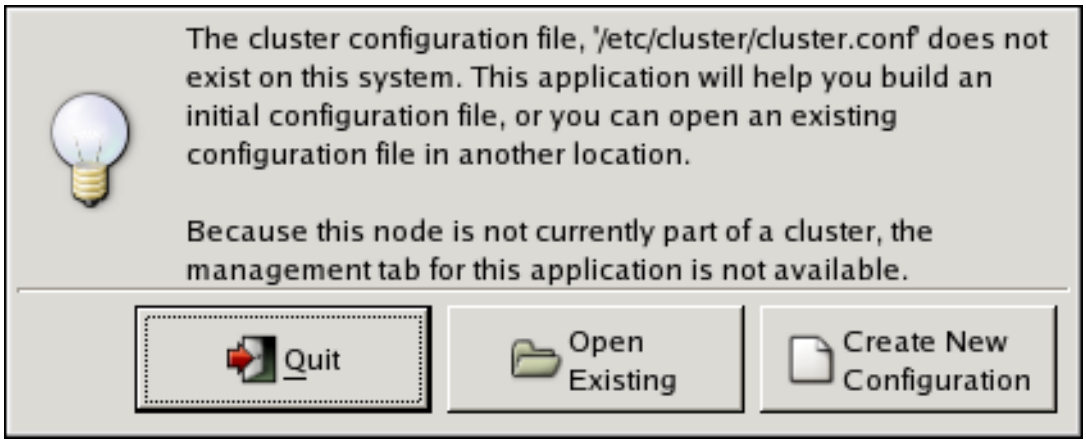

The cluster configuration file, '/etc/cluster/cluster.conf' does not exist on this system. This application will help you build an initial configuration file, or you can open an existing configuration file in another location.
Because this node is not currently part of a cluster, the management tab for this application is not available.
Quit
Open Existing
Create New Configuration

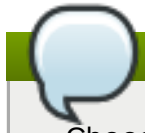
Figure 5.1. Starting a New Configuration File

Note

The **Cluster Management** tab for the Red Hat Cluster Suite management GUI is available after you save the configuration file with the **Cluster Configuration Tool**, exit, and restart the Red Hat Cluster Suite management GUI (**system-config-cluster**). (The **Cluster Management** tab displays the status of the cluster service manager, cluster nodes, and resources, and shows statistics concerning cluster service operation. To manage the cluster system further, choose the **Cluster Configuration** tab.)

3. Clicking **Create New Configuration** causes the **New Configuration** dialog box to be displayed (refer to Figure 5.2, "Creating A New Configuration"). The **New Configuration**

dialog box provides a text box for cluster name and the following checkboxes: **Custom Configure Multicast** and **Use a Quorum Disk**. In most circumstances you only need to configure the cluster name.



Note

Choose the cluster name carefully. The only way to change the name of a Red Hat cluster is to create a new cluster configuration with the new name.

Custom Configure Multicast

Red Hat Cluster software chooses a multicast address for cluster management communication among cluster nodes. If you need to use a specific multicast address, click the **Custom Configure Multicast** checkbox and enter a multicast address in the **Address** text boxes.

Note

IPV6 is not supported for Cluster Suite in Red Hat Enterprise Linux 5.

If you do not specify a multicast address, the Red Hat Cluster software (specifically, **cman**, the Cluster Manager) creates one. It forms the upper 16 bits of the multicast address with 239.192 and forms the lower 16 bits based on the cluster ID.

Note

The cluster ID is a unique identifier that **cman** generates for each cluster. To view the cluster ID, run the **cman_tool status** command on a cluster node.

If you do specify a multicast address, you should use the 239.192.x.x series that **cman** uses. Otherwise, using a multicast address outside that range may cause unpredictable results. For example, using 224.0.0.x (which is "All hosts on the network") may not be routed correctly, or even routed at all by some hardware.

Note

If you specify a multicast address, make sure that you check the configuration of routers that cluster packets pass through. Some routers may take a long time to learn addresses, seriously impacting cluster performance.

Use a Quorum Disk

If you need to use a quorum disk, click the **Use a Quorum disk** checkbox and enter quorum disk parameters. The following quorum-disk parameters are available in the dialog box if you enable **Use a Quorum disk**: **Interval**, **TKO**, **Votes**, **Minimum Score**, **Device**, **Label**, and **Quorum Disk Heuristic**. Table 5.1, “Quorum-Disk Parameters” describes the parameters.

Important

Quorum-disk parameters and heuristics depend on the site environment and special requirements needed. To understand the use of quorum-disk parameters and heuristics, refer to the qdisk(5) man page. If you require assistance understanding and using quorum disk, contact an authorized Red Hat support representative.

Note

It is probable that configuring a quorum disk requires changing quorum-disk parameters after the initial configuration. The **Cluster Configuration Tool** (**`system-config-cluster`**) provides only the display of quorum-disk parameters after initial configuration. If you need to configure quorum disk, consider using **Conga** instead; **Conga** allows modification of quorum disk parameters.

Overall:

While **`system-config-cluster`** provides several convenient tools for configuring and managing a Red Hat Cluster, the newer, more comprehensive tool, **Conga**, provides more convenience and flexibility than **`system-config-cluster`**. You may want to consider using **Conga** instead (refer to Chapter 3, *Configuring Red Hat Cluster With **Conga*** and Chapter 4, *Managing Red Hat Cluster With **Conga***).

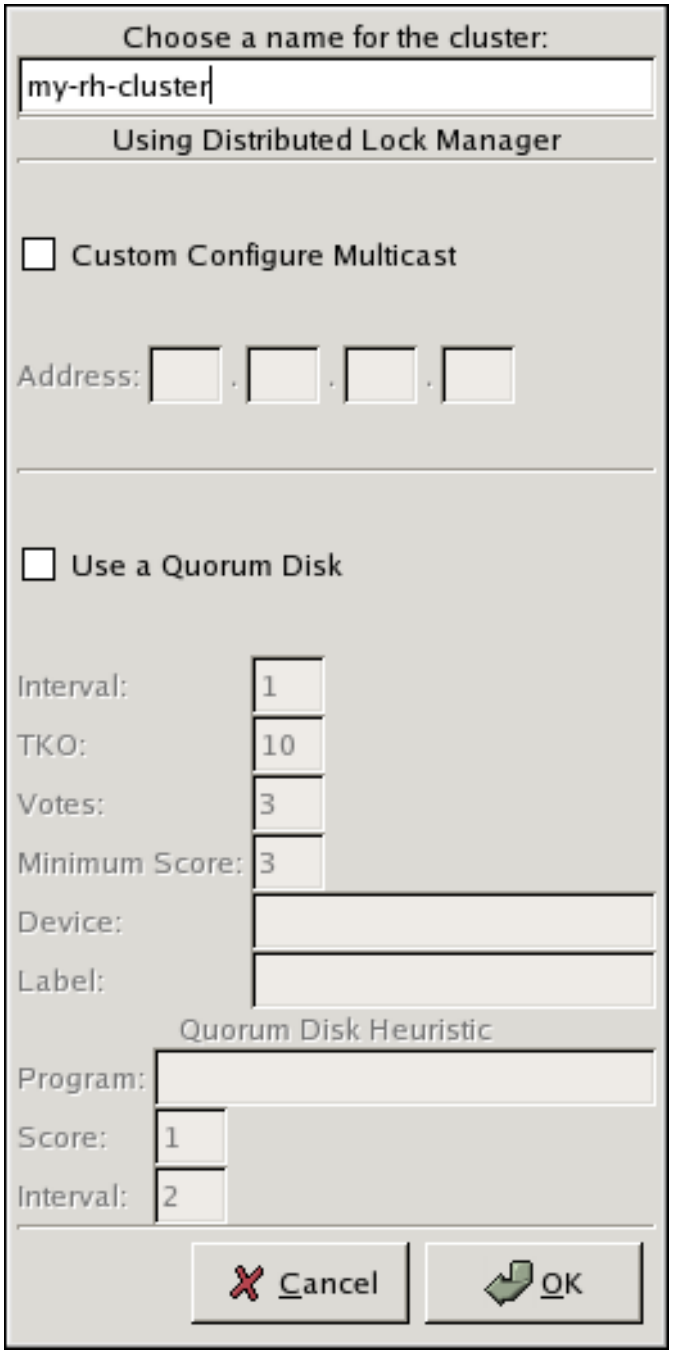


Figure 5.2. Creating A New Configuration

4. When you have completed entering the cluster name and other parameters in the **New Configuration** dialog box, click **OK**. Clicking **OK** starts the **Cluster Configuration Tool**, displaying a graphical representation of the configuration (Figure 5.3, “The Cluster Configuration Tool”).

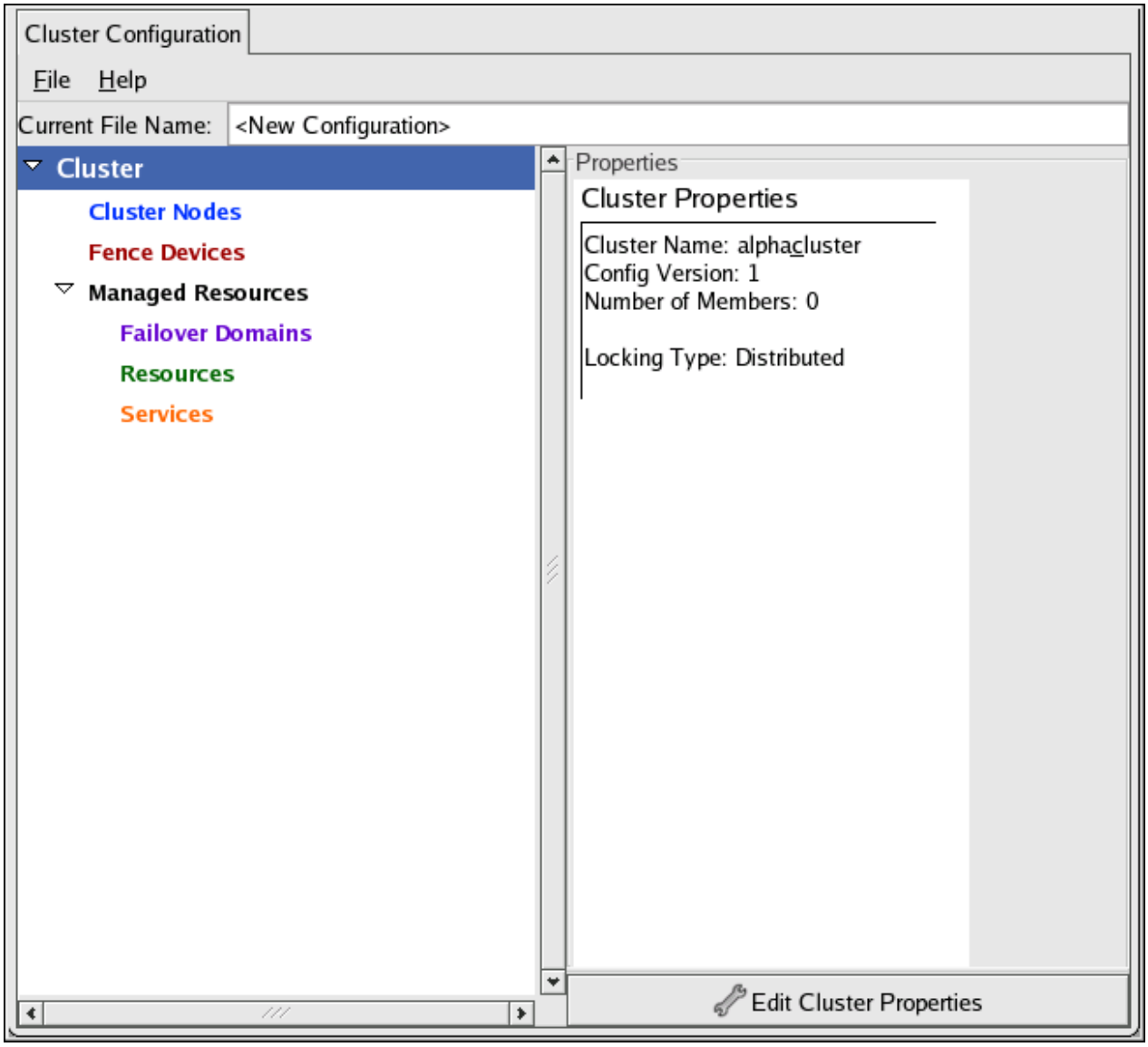


Figure 5.3. The Cluster Configuration Tool

Table 5.1. Quorum-Disk Parameters

Parameter	Description
Use a Quorum Disk	Enables quorum disk. Enables quorum-disk parameters in the **New Configuration** dialog box.
Interval	The frequency of read/write cycles, in seconds.
TKO	The number of cycles a node must miss in order to be declared dead.
Votes	The number of votes the quorum daemon advertises to CMAN when it has a high enough score.
Minimum Score	The minimum score for a node to be considered "alive". If omitted or set to 0, the default function, **floor((*n*+1)/2)**, is used, where *n* is the sum of the heuristics scores. The **Minimum Score** value must never exceed the sum of the heuristic scores; otherwise, the quorum disk cannot be available.
Device	The storage device the quorum daemon uses. The device must be the same on all nodes.

Parameter	Description
Label	Specifies the quorum disk label created by the **mkqdisk** utility. If this field contains an entry, the label overrides the **Device** field. If this field is used, the quorum daemon reads **/proc/partitions** and checks for qdisk signatures on every block device found, comparing the label against the specified label. This is useful in configurations where the quorum device name differs among nodes.
Quorum Disk Heuristics	**Program** — The program used to determine if this heuristic is alive. This can be anything that can be executed by **/bin/sh -c**. A return value of 0 indicates success; anything else indicates failure. This field is required. **Score** — The weight of this heuristic. Be careful when determining scores for heuristics. The default score for each heuristic is 1. **Interval** — The frequency (in seconds) at which the heuristic is polled. The default interval for every heuristic is 2 seconds.

5.3. Configuring Cluster Properties

In addition to configuring cluster parameters in the preceding section (Section 5.2, "Starting the **Cluster Configuration Tool**"), you can configure the following cluster properties: **Cluster Alias** (optional), a **Config Version** (optional), and **Fence Daemon Properties**. To configure cluster properties, follow these steps:

1. At the left frame, click **Cluster**.
2. At the bottom of the right frame (labeled **Properties**), click the **Edit Cluster Properties** button. Clicking that button causes a **Cluster Properties** dialog box to be displayed. The **Cluster Properties** dialog box presents text boxes for **Cluster Alias**, **Config Version**, and two **Fence Daemon Properties** parameters: **Post-Join Delay** and **Post-Fail Delay**.
3. (Optional) At the **Cluster Alias** text box, specify a cluster alias for the cluster. The default cluster alias is set to the true cluster name provided when the cluster is set up (refer to Section 5.2, "Starting the **Cluster Configuration Tool**"). The cluster alias should be descriptive enough to distinguish it from other clusters and systems on your network (for example, **nfs_cluster** or **httpd_cluster**). The cluster alias cannot exceed 15 characters.
4. (Optional) The **Config Version** value is set to **1** by default and is automatically incremented each time you save your cluster configuration. However, if you need to set it to another value, you can specify it at the **Config Version** text box.
5. Specify the **Fence Daemon Properties** parameters: **Post-Join Delay** and **Post-Fail Delay**.
    a. The **Post-Join Delay** parameter is the number of seconds the fence daemon (**fenced**) waits before fencing a node after the node joins the fence domain. The **Post-Join Delay** default value is **3**. A typical setting for **Post-Join Delay** is between 20 and 30 seconds, but can vary according to cluster and network performance.
    b. The **Post-Fail Delay** parameter is the number of seconds the fence daemon (**fenced**) waits before fencing a node (a member of the fence domain) after the node has failed.The **Post-Fail Delay** default value is **0**. Its value may be varied to suit cluster and network performance.

Note

For more information about **Post-Join Delay** and **Post-Fail Delay**, refer to the fenced(8) man page.

6. Save cluster configuration changes by selecting **File** => **Save**.

5.4. Configuring Fence Devices

Configuring fence devices for the cluster consists of selecting one or more fence devices and specifying fence-device-dependent parameters (for example, name, IP address, login, and password).

To configure fence devices, follow these steps:

1. Click **Fence Devices**. At the bottom of the right frame (labeled **Properties**), click the **Add a Fence Device** button. Clicking **Add a Fence Device** causes the **Fence Device Configuration** dialog box to be displayed (refer to Figure 5.4, "Fence Device Configuration").

Figure 5.4. Fence Device Configuration

2. At the **Fence Device Configuration** dialog box, click the drop-down box under **Add a New Fence Device** and select the type of fence device to configure.

3. Specify the information in the **Fence Device Configuration** dialog box according to the type of fence device. Refer to Appendix B, *Fence Device Parameters* for more information about fence device parameters.

4. Click **OK**.

5. Choose **File** => **Save** to save the changes to the cluster configuration.

5.5. Adding and Deleting Members

The procedure to add a member to a cluster varies depending on whether the cluster is a newly-configured cluster or a cluster that is already configured and running. To add a member to a new cluster, refer to Section 5.5.1, “Adding a Member to a Cluster”. To add a member to an existing cluster, refer to Section 5.5.2, “Adding a Member to a Running Cluster”. To delete a member from a cluster, refer to Section 5.5.3, “Deleting a Member from a Cluster”.

5.5.1. Adding a Member to a Cluster

To add a member to a new cluster, follow these steps:

1. Click **Cluster Node**.
2. At the bottom of the right frame (labeled **Properties**), click the **Add a Cluster Node** button. Clicking that button causes a **Node Properties** dialog box to be displayed. The **Node Properties** dialog box presents text boxes for **Cluster Node Name** and **Quorum Votes** (refer to Figure 5.5, “Adding a Member to a New Cluster”).

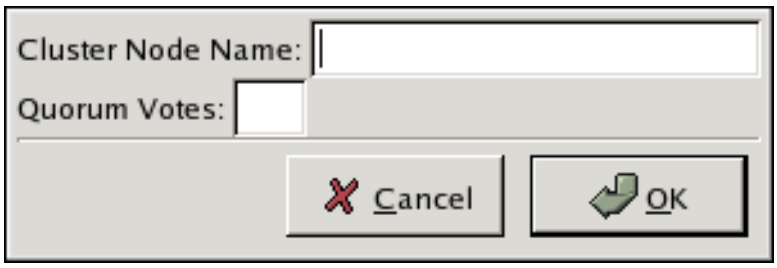


Figure 5.5. Adding a Member to a New Cluster

3. At the **Cluster Node Name** text box, specify a node name. The entry can be a name or an IP address of the node on the cluster subnet.

Note

Each node must be on the same subnet as the node from which you are running the **Cluster Configuration Tool** and must be defined either in DNS or in the **/etc/hosts** file of each cluster node.

Note

The node on which you are running the **Cluster Configuration Tool** must be explicitly added as a cluster member; the node is not automatically added to the cluster configuration as a result of running the **Cluster Configuration Tool**.

4. Optionally, at the **Quorum Votes** text box, you can specify a value; however in most configurations you can leave it blank. Leaving the **Quorum Votes** text box blank causes the quorum votes value for that node to be set to the default value of **1**.
5. Click **OK**.

6. Configure fencing for the node:
    a. Click the node that you added in the previous step.
    b. At the bottom of the right frame (below **Properties**), click **Manage Fencing For This Node**. Clicking **Manage Fencing For This Node** causes the **Fence Configuration** dialog box to be displayed.
    c. At the **Fence Configuration** dialog box, bottom of the right frame (below **Properties**), click **Add a New Fence Level**. Clicking **Add a New Fence Level** causes a fence-level element (for example, **Fence-Level-1**, **Fence-Level-2**, and so on) to be displayed below the node in the left frame of the **Fence Configuration** dialog box.
    d. Click the fence-level element.
    e. At the bottom of the right frame (below **Properties**), click **Add a New Fence to this Level**. Clicking **Add a New Fence to this Level** causes the **Fence Properties** dialog box to be displayed.
    f. At the **Fence Properties** dialog box, click the **Fence Device Type** drop-down box and select the fence device for this node. Also, provide additional information required (for example, **Port** and **Switch** for an APC Power Device).
    g. At the **Fence Properties** dialog box, click **OK**. Clicking **OK** causes a fence device element to be displayed below the fence-level element.
    h. To create additional fence devices at this fence level, return to step 6d. Otherwise, proceed to the next step.
    i. To create additional fence levels, return to step 6c. Otherwise, proceed to the next step.
    j. If you have configured all the fence levels and fence devices for this node, click **Close**.
7. Choose **File** => **Save** to save the changes to the cluster configuration.

5.5.2. Adding a Member to a Running Cluster

The procedure for adding a member to a running cluster depends on whether the cluster contains only two nodes or more than two nodes. To add a member to a running cluster, follow the steps in one of the following sections according to the number of nodes in the cluster:

- For clusters with *only* two nodes —

  Section 5.5.2.1, “Adding a Member to a Running Cluster That Contains *Only* Two Nodes”

- For clusters with *more than* two nodes —

  Section 5.5.2.2, “Adding a Member to a Running Cluster That Contains *More Than* Two Nodes”

5.5.2.1. Adding a Member to a Running Cluster That Contains *Only* Two Nodes

To add a member to an existing cluster that is currently in operation, and contains *only* two nodes, follow these steps:

1. Add the node and configure fencing for it as in

Section 5.5.1, “Adding a Member to a Cluster”.

2. Click **Send to Cluster** to propagate the updated configuration to other running nodes in the cluster.
3. Use the **scp** command to send the updated **/etc/cluster/cluster.conf** file from one of the existing cluster nodes to the new node.
4. At the Red Hat Cluster Suite management GUI **Cluster Status Tool** tab, disable each service listed under **Services**.
5. Stop the cluster software on the two running nodes by running the following commands at each node in this order:
    a. **service rgmanager stop**
    b. **service gfs stop**, if you are using Red Hat GFS
    c. **service clvmd stop**, if CLVM has been used to create clustered volumes
    d. **service cman stop**
6. Start cluster software on all cluster nodes (including the added one) by running the following commands in this order:
    a. **service cman start**
    b. **service clvmd start**, if CLVM has been used to create clustered volumes
    c. **service gfs start**, if you are using Red Hat GFS
    d. **service rgmanager start**
7. Start the Red Hat Cluster Suite management GUI. At the **Cluster Configuration Tool** tab, verify that the configuration is correct. At the **Cluster Status Tool** tab verify that the nodes and services are running as expected.

5.5.2.2. Adding a Member to a Running Cluster That Contains *More Than* Two Nodes

To add a member to an existing cluster that is currently in operation, and contains *more than* two nodes, follow these steps:

1. Add the node and configure fencing for it as in

    Section 5.5.1, “Adding a Member to a Cluster”.

2. Click **Send to Cluster** to propagate the updated configuration to other running nodes in the cluster.
3. Use the **scp** command to send the updated **/etc/cluster/cluster.conf** file from one of the existing cluster nodes to the new node.
4. Start cluster services on the new node by running the following commands in this order:
    a. **service cman start**
    b. **service clvmd start**, if CLVM has been used to create clustered volumes
    c. **service gfs start**, if you are using Red Hat GFS

d. **service rgmanager start**

5. Start the Red Hat Cluster Suite management GUI. At the **Cluster Configuration Tool** tab, verify that the configuration is correct. At the **Cluster Status Tool** tab verify that the nodes and services are running as expected.

5.5.3. Deleting a Member from a Cluster

To delete a member from an existing cluster that is currently in operation, follow these steps:

1. At one of the running nodes (not to be removed), run the Red Hat Cluster Suite management GUI. At the **Cluster Status Tool** tab, under **Services**, disable or relocate each service that is running on the node to be deleted.

2. Stop the cluster software on the node to be deleted by running the following commands at that node in this order:

   a. **service rgmanager stop**

   b. **service gfs stop**, if you are using Red Hat GFS

   c. **service clvmd stop**, if CLVM has been used to create clustered volumes

   d. **service cman stop**

3. At the **Cluster Configuration Tool** (on one of the running members), delete the member as follows:

   a. If necessary, click the triangle icon to expand the **Cluster Nodes** property.

   b. Select the cluster node to be deleted. At the bottom of the right frame (labeled **Properties**), click the **Delete Node** button.

   c. Clicking the **Delete Node** button causes a warning dialog box to be displayed requesting confirmation of the deletion (Figure 5.6, “Confirm Deleting a Member”).

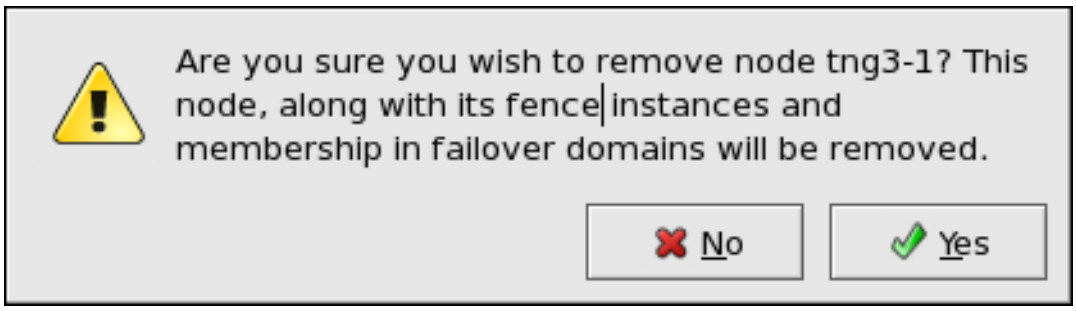

Are you sure you wish to remove node tng3-1? This node, along with its fence instances and membership in failover domains will be removed.
No
Yes

Figure 5.6. Confirm Deleting a Member

   d. At that dialog box, click **Yes** to confirm deletion.

   e. Propagate the updated configuration by clicking the **Send to Cluster** button. (Propagating the updated configuration automatically saves the configuration.)

4. Stop the cluster software on the remaining running nodes by running the following commands at each node in this order:

   a. **service rgmanager stop**

b. **service gfs stop**, if you are using Red Hat GFS

c. **service clvmd stop**, if CLVM has been used to create clustered volumes

d. **service cman stop**

5. Start cluster software on all remaining cluster nodes by running the following commands in this order:

a. **service cman start**

b. **service clvmd start**, if CLVM has been used to create clustered volumes

c. **service gfs start**, if you are using Red Hat GFS

d. **service rgmanager start**

6. Start the Red Hat Cluster Suite management GUI. At the **Cluster Configuration Tool** tab, verify that the configuration is correct. At the **Cluster Status Tool** tab verify that the nodes and services are running as expected.

5.5.3.1. Removing a Member from a Cluster at the Command-Line

If desired, you can also manually relocate and remove cluster members by using the **clusvcadm** commmand at a shell prompt.

1. To prevent service downtime, any services running on the member to be removed must be relocated to another node on the cluster by running the following command:

```
clusvcadm -r cluster_service_name -m cluster_node_name
```

Where **cluster_service_name** is the name of the service to be relocated and **cluster_member_name** is the name of the member to which the service will be relocated.

2. Stop the cluster software on the node to be removed by running the following commands at that node in this order:

a. **service rgmanager stop**

b. **service gfs stop** and/or **service gfs2 stop**, if you are using **gfs**, **gfs2** or both

c. **umount -a -t gfs** and/or **umount -a -t gfs2**, if you are using either (or both) in conjunction with **rgmanager**

d. **service clvmd stop**, if CLVM has been used to create clustered volumes

e. **service cman stop remove**

3. To ensure that the removed member does not rejoin the cluster after it reboots, run the following set of commands:

```
chkconfig cman off
chkconfig rgmanager off
chkconfig clvmd off
chkconfig gfs off
chkconfig gfs2 off
```

5.6. Configuring a Failover Domain

A failover domain is a named subset of cluster nodes that are eligible to run a cluster service in the event of a node failure. A failover domain can have the following characteristics:

- Unrestricted — Allows you to specify that a subset of members are preferred, but that a cluster service assigned to this domain can run on any available member.
- Restricted — Allows you to restrict the members that can run a particular cluster service. If none of the members in a restricted failover domain are available, the cluster service cannot be started (either manually or by the cluster software).
- Unordered — When a cluster service is assigned to an unordered failover domain, the member on which the cluster service runs is chosen from the available failover domain members with no priority ordering.
- Ordered — Allows you to specify a preference order among the members of a failover domain. The member at the top of the list is the most preferred, followed by the second member in the list, and so on.

Note

Changing a failover domain configuration has no effect on currently running services.

Note

Failover domains are *not* required for operation.

By default, failover domains are unrestricted and unordered.

In a cluster with several members, using a restricted failover domain can minimize the work to set up the cluster to run a cluster service (such as **`httpd`**), which requires you to set up the configuration identically on all members that run the cluster service). Instead of setting up the entire cluster to run the cluster service, you must set up only the members in the restricted failover domain that you associate with the cluster service.

Note

To configure a preferred member, you can create an unrestricted failover domain comprising only one cluster member. Doing that causes a cluster service to run on that cluster member primarily (the preferred member), but allows the cluster service to fail over to any of the other members.

The following sections describe adding a failover domain, removing a failover domain, and removing members from a failover domain:

- Section 5.6.1, “Adding a Failover Domain”
- Section 5.6.2, “Removing a Failover Domain”

- Section 5.6.3, “Removing a Member from a Failover Domain”

5.6.1. Adding a Failover Domain

To add a failover domain, follow these steps:

1. At the left frame of the **Cluster Configuration Tool**, click **Failover Domains**.
2. At the bottom of the right frame (labeled **Properties**), click the **Create a Failover Domain** button. Clicking the **Create a Failover Domain** button causes the **Add Failover Domain** dialog box to be displayed.
3. At the **Add Failover Domain** dialog box, specify a failover domain name at the **Name for new Failover Domain** text box and click **OK**. Clicking **OK** causes the **Failover Domain Configuration** dialog box to be displayed (Figure 5.7, “Failover Domain Configuration: Configuring a Failover Domain”).

Note

The name should be descriptive enough to distinguish its purpose relative to other names used in your cluster.

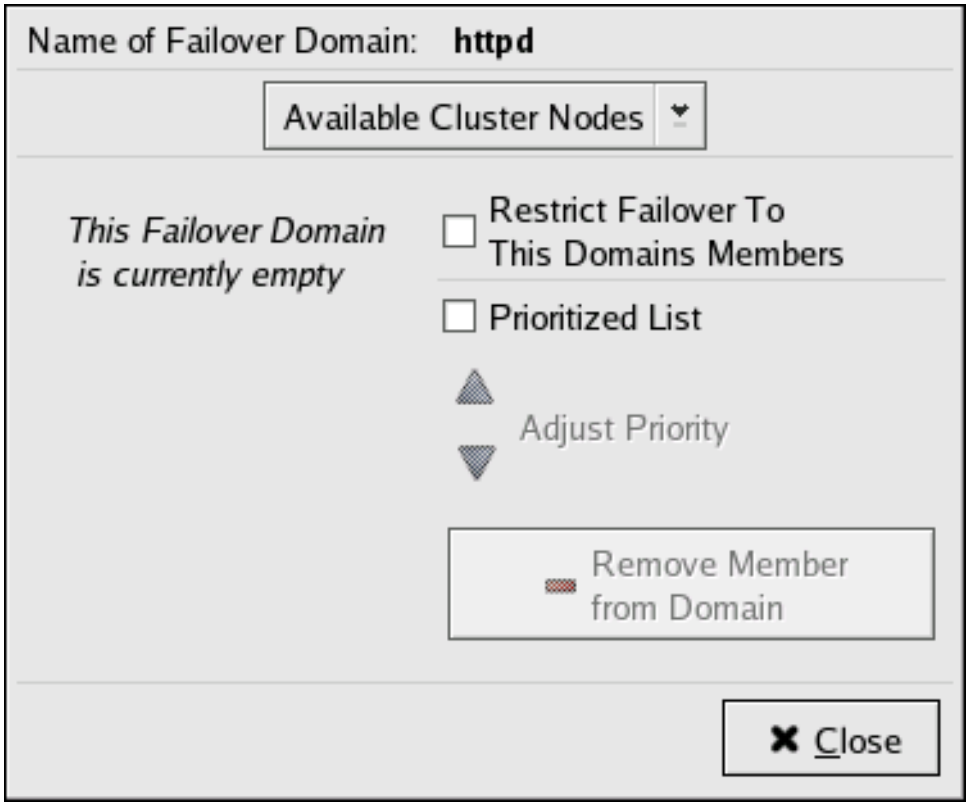


Figure 5.7. Failover Domain Configuration: Configuring a Failover Domain

4. Click the **Available Cluster Nodes** drop-down box and select the members for this failover domain.

5. To restrict failover to members in this failover domain, click (check) the **Restrict Failover To This Domains Members** checkbox. (With **Restrict Failover To This Domains Members** checked, services assigned to this failover domain fail over only to nodes in this failover domain.)

6. To prioritize the order in which the members in the failover domain assume control of a failed cluster service, follow these steps:

    a. Click (check) the **Prioritized List** checkbox (Figure 5.8, "Failover Domain Configuration: Adjusting Priority"). Clicking **Prioritized List** causes the **Priority** column to be displayed next to the **Member Node** column.

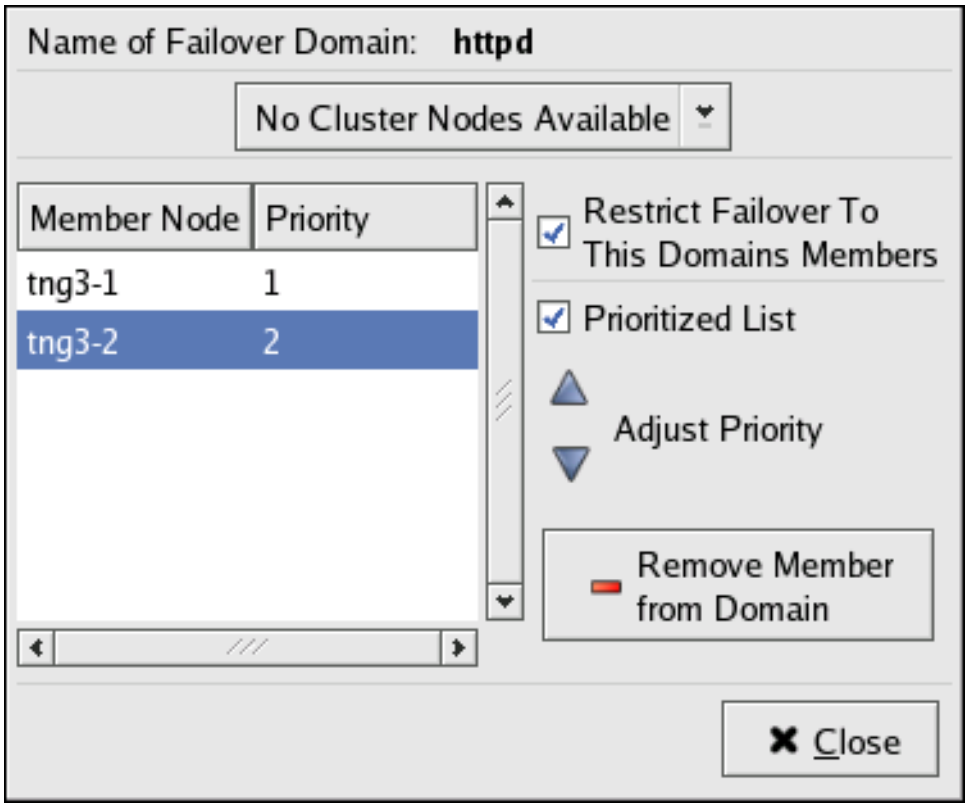


Figure 5.8. Failover Domain Configuration: Adjusting Priority

    b. For each node that requires a priority adjustment, click the node listed in the **Member Node/Priority** columns and adjust priority by clicking one of the **Adjust Priority** arrows. Priority is indicated by the position in the **Member Node** column and the value in the **Priority** column. The node priorities are listed highest to lowest, with the highest priority node at the top of the **Member Node** column (having the lowest **Priority** number).

7. Click **Close** to create the domain.

8. At the **Cluster Configuration Tool**, perform one of the following actions depending on whether the configuration is for a new cluster or for one that is operational and running:

    - New cluster — If this is a new cluster, choose **File** => **Save** to save the changes to the cluster configuration.

    - Running cluster — If this cluster is operational and running, and you want to propagate the change immediately, click the **Send to Cluster** button. Clicking **Send to Cluster** automatically saves the configuration change. If you do not want to propagate the change immediately, choose **File** => **Save** to save the changes to the cluster

configuration.

5.6.2. Removing a Failover Domain

To remove a failover domain, follow these steps:

1. At the left frame of the **Cluster Configuration Tool**, click the failover domain that you want to delete (listed under **Failover Domains**).
2. At the bottom of the right frame (labeled **Properties**), click the **Delete Failover Domain** button. Clicking the **Delete Failover Domain** button causes a warning dialog box do be displayed asking if you want to remove the failover domain. Confirm that the failover domain identified in the warning dialog box is the one you want to delete and click **Yes**. Clicking **Yes** causes the failover domain to be removed from the list of failover domains under **Failover Domains** in the left frame of the **Cluster Configuration Tool**.
3. At the **Cluster Configuration Tool**, perform one of the following actions depending on whether the configuration is for a new cluster or for one that is operational and running:
   - New cluster — If this is a new cluster, choose **File** => **Save** to save the changes to the cluster configuration.
   - Running cluster — If this cluster is operational and running, and you want to propagate the change immediately, click the **Send to Cluster** button. Clicking **Send to Cluster** automatically saves the configuration change. If you do not want to propagate the change immediately, choose **File** => **Save** to save the changes to the cluster configuration.

5.6.3. Removing a Member from a Failover Domain

To remove a member from a failover domain, follow these steps:

1. At the left frame of the **Cluster Configuration Tool**, click the failover domain that you want to change (listed under **Failover Domains**).
2. At the bottom of the right frame (labeled **Properties**), click the **Edit Failover Domain Properties** button. Clicking the **Edit Failover Domain Properties** button causes the **Failover Domain Configuration** dialog box to be displayed (Figure 5.7, “Failover Domain Configuration: Configuring a Failover Domain”).
3. At the **Failover Domain Configuration** dialog box, in the **Member Node** column, click the node name that you want to delete from the failover domain and click the **Remove Member from Domain** button. Clicking **Remove Member from Domain** removes the node from the **Member Node** column. Repeat this step for each node that is to be deleted from the failover domain. (Nodes must be deleted one at a time.)
4. When finished, click **Close**.
5. At the **Cluster Configuration Tool**, perform one of the following actions depending on whether the configuration is for a new cluster or for one that is operational and running:
   - New cluster — If this is a new cluster, choose **File** => **Save** to save the changes to the cluster configuration.
   - Running cluster — If this cluster is operational and running, and you want to propagate the change immediately, click the **Send to Cluster** button. Clicking **Send to Cluster** automatically saves the configuration change. If you do not want to propagate

the change immediately, choose **File** => **Save** to save the changes to the cluster configuration.

5.7. Adding Cluster Resources

To specify a resource for a cluster service, follow these steps:

1. On the **Resources** property of the **Cluster Configuration Tool**, click the **Create a Resource** button. Clicking the **Create a Resource** button causes the **Resource Configuration** dialog box to be displayed.
2. At the **Resource Configuration** dialog box, under **Select a Resource Type**, click the drop-down box. At the drop-down box, select a resource to configure. Appendix C, *HA Resource Parameters* describes resource parameters.
3. When finished, click **OK**.
4. Choose **File** => **Save** to save the change to the **/etc/cluster/cluster.conf** configuration file.

5.8. Adding a Cluster Service to the Cluster

To add a cluster service to the cluster, follow these steps:

1. At the left frame, click **Services**.
2. At the bottom of the right frame (labeled **Properties**), click the **Create a Service** button. Clicking **Create a Service** causes the **Add a Service** dialog box to be displayed.
3. At the **Add a Service** dialog box, type the name of the service in the **Name** text box and click **OK**. Clicking **OK** causes the **Service Management** dialog box to be displayed (refer to Figure 5.9, “Adding a Cluster Service”).

Note

Use a descriptive name that clearly distinguishes the service from other services in the cluster.

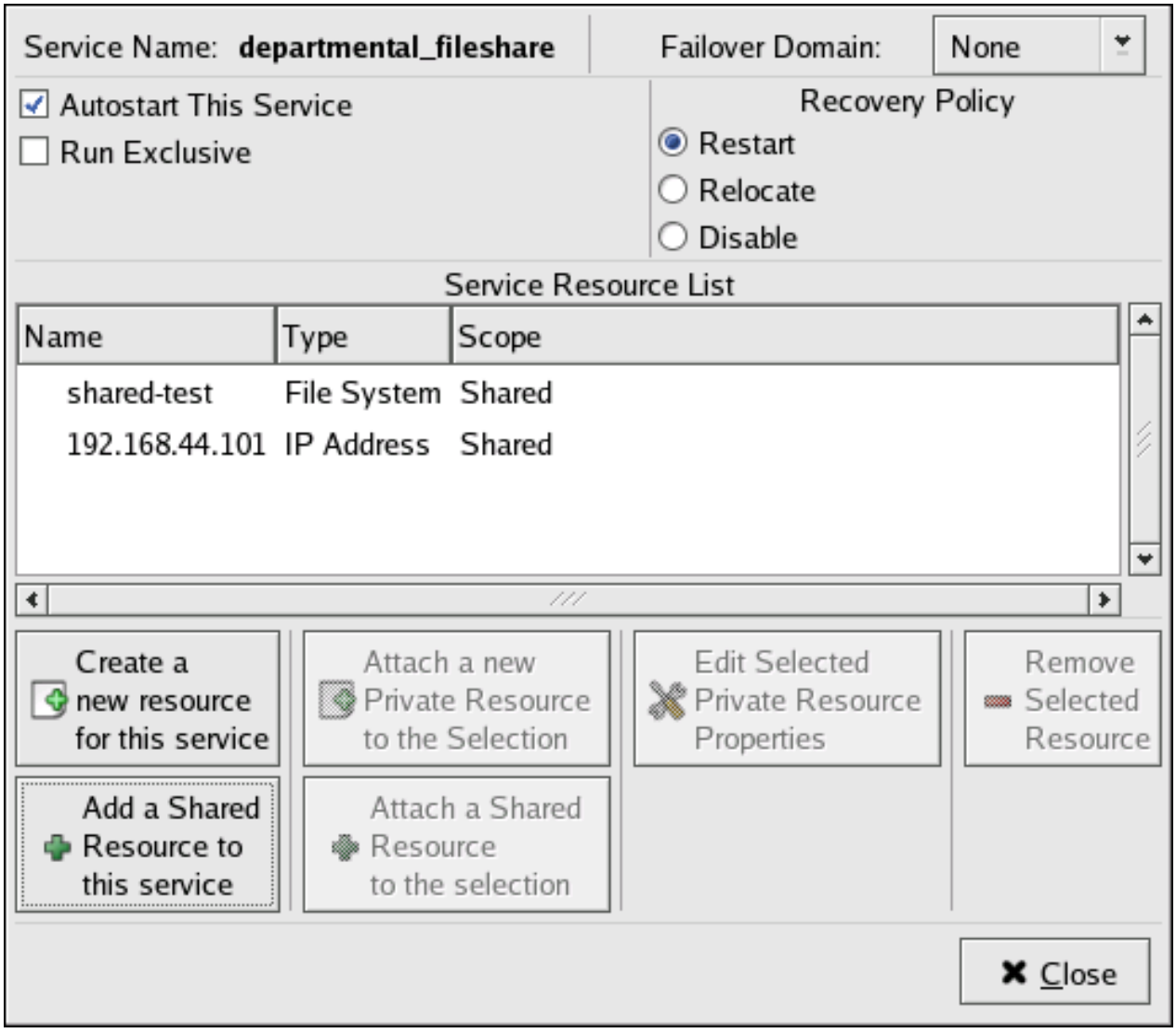


Figure 5.9. Adding a Cluster Service

4. If you want to restrict the members on which this cluster service is able to run, choose a failover domain from the **Failover Domain** drop-down box. (Refer to Section 5.6, "Configuring a Failover Domain" for instructions on how to configure a failover domain.)

5. **Autostart This Service** checkbox — This is checked by default. If **Autostart This Service** is checked, the service is started automatically when a cluster is started and running. If **Autostart This Service** is *not* checked, the service must be started manually any time the cluster comes up from stopped state.

6. **Run Exclusive** checkbox — This sets a policy wherein the service only runs on nodes that have *no other* services running on them. For example, for a very busy web server that is clustered for high availability, it would would be advisable to keep that service on a node alone with no other services competing for his resources — that is, **Run Exclusive** checked. On the other hand, services that consume few resources (like NFS and Samba), can run together on the same node without little concern over contention for resources. For those types of services you can leave the **Run Exclusive** unchecked.

Note

Circumstances that require enabling **Run Exclusive** are rare. Enabling **Run Exclusive** can render a service offline if the node it is running on fails and no other nodes are empty.

7. Select a recovery policy to specify how the resource manager should recover from a service failure. At the upper right of the **Service Management** dialog box, there are three **Recovery Policy** options available:

   - **Restart** — Restart the service in the node the service is currently located. The default setting is **Restart**. If the service cannot be restarted in the current node, the service is relocated.
   - **Relocate** — Relocate the service before restarting. Do not restart the node where the service is currently located.
   - **Disable** — Do not restart the service at all.

8. Click the **Add a Shared Resource to this service** button and choose the a resource listed that you have configured in Section 5.7, “Adding Cluster Resources”.

Note

If you are adding a Samba-service resource, connect a Samba-service resource directly to the service, *not* to a resource within a service. That is, at the **Service Management** dialog box, use either **Create a new resource for this service** or **Add a Shared Resource to this service**; do *not* use **Attach a new Private Resource to the Selection** or **Attach a Shared Resource to the selection**.

9. If needed, you may also create a *private* resource that you can create that becomes a subordinate resource by clicking on the **Attach a new Private Resource to the Selection** button. The process is the same as creating a shared resource described in Section 5.7, “Adding Cluster Resources”. The private resource will appear as a child to the shared resource to which you associated with the shared resource. Click the triangle icon next to the shared resource to display any private resources associated.

10. When finished, click **OK**.

11. Choose **File** => **Save** to save the changes to the cluster configuration.

Note

To verify the existence of the IP service resource used in a cluster service, you must use the **/sbin/ip addr list** command on a cluster node. The following output shows the **/sbin/ip addr list** command executed on a node running a cluster service:

```
1: lo: <LOOPBACK,UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP> mtu 1356 qdisc pfifo_fast qlen 1000
    link/ether 00:05:5d:9a:d8:91 brd ff:ff:ff:ff:ff:ff
    inet 10.11.4.31/22 brd 10.11.7.255 scope global eth0
    inet6 fe80::205:5dff:fe9a:d891/64 scope link
    inet 10.11.4.240/22 scope global secondary eth0
       valid_lft forever preferred_lft forever
```

5.8.1. Relocating a Service in a Cluster

Service relocation functionality allows you to perform maintenance on a cluster member while maintaining application and data availability.

To relocate a service, drag the service icon from the **Services** Tab onto the member icon in the **Members** tab. The cluster manager stops the service on the member on which it was running and restarts it on the new member.

5.9. Propagating The Configuration File: New Cluster

For newly defined clusters, you must propagate the configuration file to the cluster nodes as follows:

1. Log in to the node where you created the configuration file.
2. Using the **scp** command, copy the **/etc/cluster/cluster.conf** file to all nodes in the cluster.

Note

Propagating the cluster configuration file this way is necessary for the first time a cluster is created. Once a cluster is installed and running, the cluster configuration file is propagated using the Red Hat cluster management GUI **Send to Cluster** button. For more information about propagating the cluster configuration using the GUI **Send to Cluster** button, refer to Section 6.3, “Modifying the Cluster Configuration”.

5.10. Starting the Cluster Software

After you have propagated the cluster configuration to the cluster nodes you can either reboot each node or start the cluster software on each cluster node by running the following commands at each node in this order:

1. **`service cman start`**
2. **`service clvmd start`**, if CLVM has been used to create clustered volumes

Note

Shared storage for use in Red Hat Cluster Suite requires that you be running the cluster logical volume manager daemon (**`clvmd`**) or the High Availability Logical Volume Management agents (HA-LVM). If you are not able to use either the **`clvmd`** daemon or HA-LVM for operational reasons or because you do not have the correct entitlements, you must not use single-instance LVM on the shared disk as this may result in data corruption. If you have any concerns please contact your Red Hat service representative.

3. **`service gfs start`**, if you are using Red Hat GFS
4. **`service rgmanager start`**
5. Start the Red Hat Cluster Suite management GUI. At the **Cluster Configuration Tool** tab, verify that the configuration is correct. At the **Cluster Status Tool** tab verify that the nodes and services are running as expected.

Chapter 6. Managing Red Hat Cluster With system-config-cluster

This chapter describes various administrative tasks for managing a Red Hat Cluster and consists of the following sections:

- Section 6.1, “Starting and Stopping the Cluster Software”
- Section 6.2, “Managing High-Availability Services”
- Section 6.4, “Backing Up and Restoring the Cluster Database”
- Section 6.6, “Disabling the Cluster Software”
- Section 6.7, “Diagnosing and Correcting Problems in a Cluster”

Note

While **system-config-cluster** provides several convenient tools for configuring and managing a Red Hat Cluster, the newer, more comprehensive tool, **Conga**, provides more convenience and flexibility than **system-config-cluster**. You may want to consider using **Conga** instead (refer to Chapter 3, *Configuring Red Hat Cluster With **Conga*** and Chapter 4, *Managing Red Hat Cluster With **Conga***).

6.1. Starting and Stopping the Cluster Software

To start the cluster software on a member, type the following commands in this order:

1. **service cman start**
2. **service clvmd start**, if CLVM has been used to create clustered volumes
3. **service gfs start**, if you are using Red Hat GFS
4. **service rgmanager start**

To stop the cluster software on a member, type the following commands in this order:

1. **service rgmanager stop**
2. **service gfs stop**, if you are using Red Hat GFS
3. **service clvmd stop**, if CLVM has been used to create clustered volumes
4. **service cman stop**

Stopping the cluster services on a member causes its services to fail over to an active member.

6.2. Managing High-Availability Services

You can manage cluster services with the **Cluster Status Tool** (Figure 6.1, “Cluster Status Tool”) through the **Cluster Management** tab in Cluster Administration GUI.

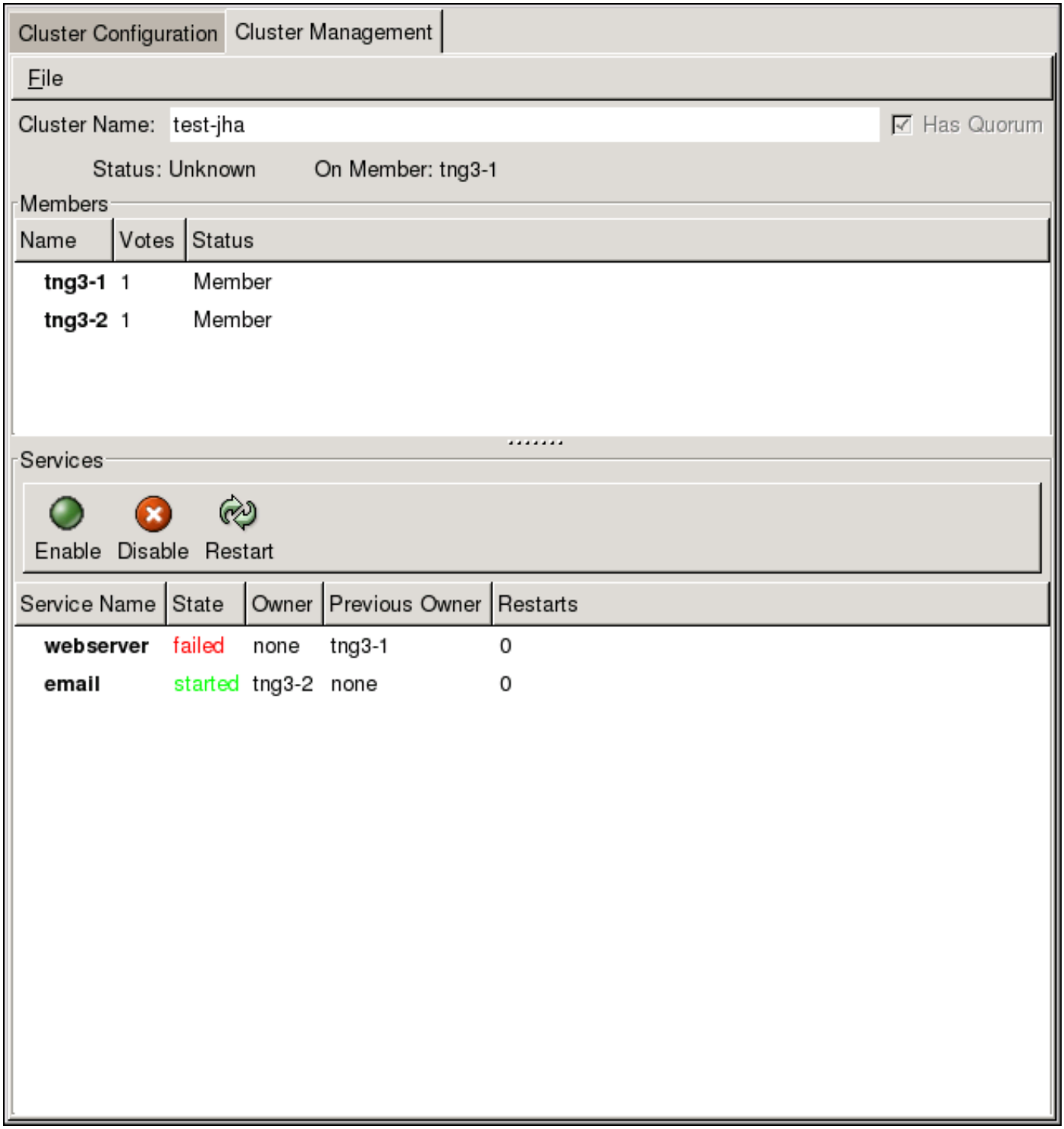


Figure 6.1. Cluster Status Tool

You can use the **Cluster Status Tool** to enable, disable, restart, or relocate a high-availability service. The **Cluster Status Tool** displays the current cluster status in the **Services** area and automatically updates the status every 10 seconds.

To enable a service, you can select the service in the **Services** area and click **Enable**. To disable a service, you can select the service in the **Services** area and click **Disable**. To restart a service, you can select the service in the **Services** area and click **Restart**. To relocate a service from one node to another, you can drag the service to another node and drop the service onto that node. Relocating a service restarts the service on that node. (Relocating a service to its current node — that is, dragging a service to its current node and dropping the service onto that node — restarts the service.)

The following tables describe the members and services status information displayed by the **Cluster Status Tool**.

Table 6.1. Members Status

Members Status	Description
Member	The node is part of the cluster. Note: A node can be a member of a cluster; however, the node may be inactive and incapable of running services. For example, if **rgmanager** is not running on the node, but all other cluster software components are running in the node, the node appears as a **Member** in the **Cluster Status Tool**.
Dead	The node is unable to participate as a cluster member. The most basic cluster software is not running on the node.

Table 6.2. Services Status

Services Status	Description
Started	The service resources are configured and available on the cluster system that owns the service.
Pending	The service has failed on a member and is pending start on another member.
Disabled	The service has been disabled, and does not have an assigned owner. A disabled service is never restarted automatically by the cluster.
Stopped	The service is not running; it is waiting for a member capable of starting the service. A service remains in the stopped state if autostart is disabled.
Failed	The service has failed to start on the cluster and cannot successfully stop the service. A failed service is never restarted automatically by the cluster.

6.3. Modifying the Cluster Configuration

To modify the cluster configuration (the cluster configuration file (**/etc/cluster/cluster.conf**), use the **Cluster Configuration Tool**. For more information about using the **Cluster Configuration Tool**, refer to Chapter 5, *Configuring Red Hat Cluster With* ***system-config-cluster***.

Warning

Do not manually edit the contents of the **/etc/cluster/cluster.conf** file without guidance from an authorized Red Hat representative or unless you fully understand the consequences of editing the **/etc/cluster/cluster.conf** file manually.

Important

Although the **Cluster Configuration Tool** provides a **Quorum Votes** parameter in the **Properties** dialog box of each cluster member, that parameter is intended *only* for use during initial cluster configuration. Furthermore, it is recommended that you retain the default **Quorum Votes** value of **1**. For more information about using the **Cluster Configuration Tool**, refer to Chapter 5, *Configuring Red Hat Cluster With* ***system-config-cluster***.

To edit the cluster configuration file, click the **Cluster Configuration** tab in the cluster configuration GUI. Clicking the **Cluster Configuration** tab displays a graphical representation of the cluster configuration. Change the configuration file according the following steps:

1. Make changes to cluster elements (for example, create a service).
2. Propagate the updated configuration file throughout the cluster by clicking **Send to Cluster**.

Note

The **Cluster Configuration Tool** does not display the **Send to Cluster** button if the cluster is new and has not been started yet, or if the node from which you are running the **Cluster Configuration Tool** is not a member of the cluster. If the **Send to Cluster** button is not displayed, you can still use the **Cluster Configuration Tool**; however, you cannot propagate the configuration. You can still *save* the configuration file. For information about using the **Cluster Configuration Tool** for a new cluster configuration, refer to Chapter 5, *Configuring Red Hat Cluster With* ***system-config-cluster***.

3. Clicking **Send to Cluster** causes a **Warning** dialog box to be displayed. Click **Yes** to save and propagate the configuration.
4. Clicking **Yes** causes an **Information** dialog box to be displayed, confirming that the current configuration has been propagated to the cluster. Click **OK**.
5. Click the **Cluster Management** tab and verify that the changes have been propagated to the cluster members.

6.4. Backing Up and Restoring the Cluster Database

The **Cluster Configuration Tool** automatically retains backup copies of the three most recently used configuration files (besides the currently used configuration file). Retaining the backup copies is useful if the cluster does not function correctly because of misconfiguration and you need to return to a previous working configuration.

Each time you save a configuration file, the **Cluster Configuration Tool** saves backup copies of the three most recently used configuration files as **/etc/cluster/cluster.conf.bak.1**, **/etc/cluster/cluster.conf.bak.2**, and **/etc/cluster/cluster.conf.bak.3**. The backup file **/etc/cluster/cluster.conf.bak.1** is the newest backup, **/etc/cluster/cluster.conf.bak.2** is the second newest backup, and **/etc/cluster/cluster.conf.bak.3** is the third newest backup.

If a cluster member becomes inoperable because of misconfiguration, restore the configuration file according to the following steps:

1. At the **Cluster Configuration Tool** tab of the Red Hat Cluster Suite management GUI, click **File** => **Open**.
2. Clicking **File => Open** causes the **system-config-cluster** dialog box to be displayed.
3. At the **system-config-cluster** dialog box, select a backup file (for example, **/etc/cluster/cluster.conf.bak.1**). Verify the file selection in the **Selection** box and click **OK**.
4. Increment the configuration version beyond the current working version number as follows:
    a. Click **Cluster** => **Edit Cluster Properties**.
    b. At the **Cluster Properties** dialog box, change the **Config Version** value and click **OK**.
5. Click **File** => **Save As**.
6. Clicking **File** => **Save As** causes the **system-config-cluster** dialog box to be displayed.
7. At the **system-config-cluster** dialog box, select **/etc/cluster/cluster.conf** and click **OK**. (Verify the file selection in the **Selection** box.)
8. Clicking **OK** causes an **Information** dialog box to be displayed. At that dialog box, click **OK**.
9. Propagate the updated configuration file throughout the cluster by clicking **Send to Cluster**.

Note

The **Cluster Configuration Tool** does not display the **Send to Cluster** button if the cluster is new and has not been started yet, or if the node from which you are running the **Cluster Configuration Tool** is not a member of the cluster. If the **Send to Cluster** button is not displayed, you can still use the **Cluster Configuration Tool**; however, you cannot propagate the configuration. You can still *save* the configuration file. For information about using the **Cluster Configuration Tool** for a new cluster configuration, refer to Chapter 5, *Configuring Red Hat Cluster With* ***system-config-cluster***.

10. Clicking **Send to Cluster** causes a **Warning** dialog box to be displayed. Click **Yes** to propagate the configuration.
11. Click the **Cluster Management** tab and verify that the changes have been propagated to the cluster members.

6.5. Disabling Resources of a Clustered Service for Maintenance

At times, it may be necessary to to stop a resource that is part of a clustered service. You can configure services in the **cluster.conf** file to have hierarchical resources (similar to a dependency tree) to disable a resource in a service without disabling other resources within that service.

So, for example, if you have a database that uses an ext3-formatted filesystem, you can disable the database while preserving the filesystem resource for use in the service.

In the following example snippet of a **cluster.conf** file, a service uses a MySQL database and ext3-formatted filesystem resources.

```
<resources>
     <mysql config_file="/etc/my.cnf" name="mysql-resource" shutdown_wait="0"/>
     <fs device="/dev/sdb1" force_fsck="0" force_unmount="1" fsid="9349"
fstype="ext3" mountpoint="/opt/db" name="SharedDisk" self_fence="0"/>
</resources>

<service name="ha-mysql">
     <fs ref="SharedDisk">
          <mysql ref="mysql-resource"/>
     </fs>
</service>
```

In order to stop the MySQL-database and perform maintenance tasks without interfering with the cluster software (mainly rgmanager), you must first freeze the clustered service:

```
clusvcadm -Z ha-mysql
```

You can then stop the MySQL service with the **rg_test** command:

```
rg_test test /etc/cluster/cluster.conf stop mysql mysql-resource
```

When the MySQL database has been shutdown, maintenance can be performed. After finishing the maintenance, start the MySQL database with rg_test again:

```
rg_test test /etc/cluster/cluster.conf start mysql mysql-resource
```

The cluster service is still frozen and will not be monitored by rgmanager. To enable monitoring again, unfreeze the clustered service:

```
clusvcadm -U ha-mysql
```

Note

The **rg_test** utility will stop all instances of a resource on a given node, potentially causing undesired results if multiple services on a single node are sharing the same resource. Do not perform these steps on resources that have multiple instances within the **cluster.conf** file. In such cases, it is usually necessary to disable the service for maintenance.

6.6. Disabling the Cluster Software

It may become necessary to temporarily disable the cluster software on a cluster member. For example, if a cluster member experiences a hardware failure, you may want to reboot that member, but prevent it from rejoining the cluster to perform maintenance on the system.

Use the **/sbin/chkconfig** command to stop the member from joining the cluster at boot-up as follows:

```
# chkconfig --level 2345 rgmanager off
# chkconfig --level 2345 gfs off
# chkconfig --level 2345 clvmd off
# chkconfig --level 2345 cman off
```

Once the problems with the disabled cluster member have been resolved, use the following commands to allow the member to rejoin the cluster:

```
# chkconfig --level 2345 rgmanager on
# chkconfig --level 2345 gfs on
# chkconfig --level 2345 clvmd on
# chkconfig --level 2345 cman on
```

You can then reboot the member for the changes to take effect or run the following commands in the order shown to restart cluster software:

1. **service cman start**
2. **service clvmd start**, if CLVM has been used to create clustered volumes
3. **service gfs start**, if you are using Red Hat GFS
4. **service rgmanager start**

6.7. Diagnosing and Correcting Problems in a Cluster

For information about diagnosing and correcting problems in a cluster, contact an authorized Red Hat support representative.

Example of Setting Up Apache HTTP Server

This appendix provides an example of setting up a highly available Apache HTTP Server on a Red Hat Cluster. The example describes how to set up a service to fail over an Apache HTTP Server. Variables in the example apply to this example only; they are provided to assist setting up a service that suits your requirements.

Note

This example uses the **Cluster Configuration Tool** (**system-config-cluster**). You can use comparable **Conga** functions to make an Apache HTTP Server highly available on a Red Hat Cluster.

A.1. Apache HTTP Server Setup Overview

First, configure Apache HTTP Server on all nodes in the cluster. If using a failover domain , assign the service to all cluster nodes configured to run the Apache HTTP Server. Refer to Section 5.6, "Configuring a Failover Domain" for instructions. The cluster software ensures that only one cluster system runs the Apache HTTP Server at one time. The example configuration consists of installing the **httpd** RPM package on all cluster nodes (or on nodes in the failover domain, if used) and configuring a shared GFS shared resource for the Web content.

When installing the Apache HTTP Server on the cluster systems, run the following command to ensure that the cluster nodes do not automatically start the service when the system boots:

```
# chkconfig --del httpd
```

Rather than having the system init scripts spawn the **httpd** daemon, the cluster infrastructure initializes the service on the active cluster node. This ensures that the corresponding IP address and file system mounts are active on only one cluster node at a time.

When adding an **httpd** service, a *floating* IP address must be assigned to the service so that the IP address will transfer from one cluster node to another in the event of failover or service relocation. The cluster infrastructure binds this IP address to the network interface on the cluster system that is currently running the Apache HTTP Server. This IP address ensures that the cluster node running **httpd** is transparent to the clients accessing the service.

The file systems that contain the Web content cannot be automatically mounted on the shared storage resource when the cluster nodes boot. Instead, the cluster software must mount and unmount the file system as the **httpd** service is started and stopped. This prevents the cluster systems from accessing the same data simultaneously, which may result in data corruption. Therefore, do not include the file systems in the **/etc/fstab** file.

A.2. Configuring Shared Storage

To set up the shared file system resource, perform the following tasks as root on one cluster system:

1. On one cluster node, use the interactive **parted** utility to create a partition to use for the document root directory. Note that it is possible to create multiple document root directories on different disk partitions.

2. Use the **mkfs** command to create an ext3 file system on the partition you created in the previous step. Specify the drive letter and the partition number. For example:

```
# mkfs -t ext3 /dev/sde3
```

3. Mount the file system that contains the document root directory. For example:

```
# mount /dev/sde3 /var/www/html
```

   Do not add this mount information to the **/etc/fstab** file because only the cluster software can mount and unmount file systems used in a service.

4. Copy all the required files to the document root directory.

5. If you have CGI files or other files that must be in different directories or in separate partitions, repeat these steps, as needed.

A.3. Installing and Configuring the Apache HTTP Server

The Apache HTTP Server must be installed and configured on all nodes in the assigned failover domain, if used, or in the cluster. The basic server configuration must be the same on all nodes on which it runs for the service to fail over correctly. The following example shows a basic Apache HTTP Server installation that includes no third-party modules or performance tuning.

On all node in the cluster (or nodes in the failover domain, if used), install the **httpd** RPM package. For example:

rpm -Uvh httpd-<*version*>.<*arch*>.rpm

To configure the Apache HTTP Server as a cluster service, perform the following tasks:

1. Edit the **/etc/httpd/conf/httpd.conf** configuration file and customize the file according to your configuration. For example:

   - Specify the directory that contains the HTML files. Also specify this mount point when adding the service to the cluster configuration. It is only required to change this field if the mount point for the web site's content differs from the default setting of **/var/www/html/**. For example:

     ```
     DocumentRoot "/mnt/httpdservice/html"
     ```

   - Specify a unique IP address to which the service will listen for requests. For example:

     ```
     Listen 192.168.1.100:80
     ```

     This IP address then must be configured as a cluster resource for the service using the **Cluster Configuration Tool**.

   - If the script directory resides in a non-standard location, specify the directory that contains the CGI programs. For example:

     ```
     ScriptAlias /cgi-bin/ "/mnt/httpdservice/cgi-bin/"
     ```

   - Specify the path that was used in the previous step, and set the access permissions to default to that directory. For example:

```
<Directory /mnt/httpdservice/cgi-bin">
AllowOverride None
Options None
Order allow,deny
Allow from all
</Directory>
```

Additional changes may need to be made to tune the Apache HTTP Server or add module functionality. For information on setting up other options, refer to the *Red Hat Enterprise Linux System Administration Guide* and the *Red Hat Enterprise Linux Reference Guide*.

2. The standard Apache HTTP Server start script, **/etc/rc.d/init.d/httpd** is also used within the cluster framework to start and stop the Apache HTTP Server on the active cluster node. Accordingly, when configuring the service, specify this script by adding it as a **Script** resource in the **Cluster Configuration Tool**.
3. Copy the configuration file over to the other nodes of the cluster (or nodes of the failover domain, if configured).

Before the service is added to the cluster configuration, ensure that the Apache HTTP Server directories are not mounted. Then, on one node, invoke the **Cluster Configuration Tool** to add the service, as follows. This example assumes a failover domain named **httpd-domain** was created for this service.

1. Add the init script for the Apache HTTP Server service.
   - Select the **Resources** tab and click **Create a Resource**. The **Resources Configuration** properties dialog box is displayed.
   - Select **Script** form the drop down menu.
   - Enter a **Name** to be associated with the Apache HTTP Server service.
   - Specify the path to the Apache HTTP Server init script (for example, **/etc/rc.d/init.d/httpd**) in the **File (with path)** field.
   - Click **OK**.
2. Add a device for the Apache HTTP Server content files and/or custom scripts.
   - Click **Create a Resource**.
   - In the **Resource Configuration** dialog, select **File System** from the drop-down menu.
   - Enter the **Name** for the resource (for example, **httpd-content**.
   - Choose **ext3** from the **File System Type** drop-down menu.
   - Enter the mount point in the **Mount Point** field (for example, **/var/www/html/**).
   - Enter the device special file name in the **Device** field (for example, **/dev/sda3**).
3. Add an IP address for the Apache HTTP Server service.
   - Click **Create a Resource**.
   - Choose **IP Address** from the drop-down menu.
   - Enter the **IP Address** to be associated with the Apache HTTP Server service.

- Make sure that the **Monitor Link** checkbox is left checked.
- Click **OK**.

4. Click the **Services** property.
5. Create the Apache HTTP Server service.
   - Click **Create a Service**. Type a **Name** for the service in the **Add a Service** dialog.
   - In the **Service Management** dialog, select a **Failover Domain** from the drop-down menu or leave it as **None**.
   - Click the **Add a Shared Resource to this service** button. From the available list, choose each resource that you created in the previous steps. Repeat this step until all resources have been added.
   - Click **OK**.
6. Choose **File** => **Save** to save your changes.

Fence Device Parameters

This appendix provides tables with parameter descriptions of fence devices.

Note

The **Name** parameter for a fence device specifies an arbitrary name for the device that will be used by Red Hat Cluster Suite. This is not the same as the DNS name for the device.

Note

Certain fence devices have an optional **Password Script** parameter. The **Password Script** parameter allows specifying that a fence-device password is supplied from a script rather than from the **Password** parameter. Using the **Password Script** parameter supersedes the **Password** parameter, allowing passwords to not be visible in the cluster configuration file (**`/etc/cluster/cluster.conf`**).

Table B.1. APC Power Switch

Field	Description
Name	A name for the APC device connected to the cluster.
IP Address	The IP address assigned to the device.
Login	The login name used to access the device.
Password	The password used to authenticate the connection to the device.
Password Script (optional)	The script that supplies a password for access to the fence device. Using this supersedes the **Password** parameter.
Power wait	Number of seconds to wait after issuing a power off or power on command.
Port	The port.
Switch (optional)	The switch number for the APC switch that connects to the node when you have multiple daisy-chained switches.
Use SSH	(Red Hat Enterprise Linux 5.4 and later) Indicates that system will use SSH to access the device.
`fence_apc`	The fence agent for APC.

Table B.2. APC Power Switch over SNMP (Red Hat Enterprise Linux 5.2 and later)

Field	Description
Name	A name for the APC device connected to the cluster into which the fence daemon logs via the SNMP protocol.
IP Address	The IP address or hostname assigned to the device.
UDP/TCP Port	The UDP/TCP port to use for connection with the device; the default value is 161.
Login	The login name used to access the device.
Password	The password used to authenticate the connection to the device.
Password Script (optional)	The script that supplies a password for access to the fence device. Using this supersedes the **Password** parameter.

Field	Description
SNMP version	The SNMP version to use (1, 2c, 3); the default value is 1.
SNMP community	The SNMP community string; the default value is **private**.
SNMP security level	The SNMP security level (noAuthNoPriv, authNoPriv, authPriv).
SNMP authentication protocol	The SNMP authentication protocol (MD5, SHA).
SNMP privacy protocol	The SNMP privacy protocol (DES, AES).
SNMP privacy protocol password	The SNMP privacy protocol password.
SNMP privacy protocol script	The script that supplies a password for SNMP privacy protocol. Using this supersedes the **SNMP privacy protocol password** parameter.
Power wait	Number of seconds to wait after issuing a power off or power on command.
Port	The port.
fence_apc_snmp	The fence agent for APC that logs into the SNP device via the SNMP protocol.

Table B.3. Brocade Fabric Switch

Field	Description
Name	A name for the Brocade device connected to the cluster.
IP Address	The IP address assigned to the device.
Login	The login name used to access the device.
Password	The password used to authenticate the connection to the device.
Password Script (optional)	The script that supplies a password for access to the fence device. Using this supersedes the **Password** parameter.
Port	The switch outlet number.
fence_brocade	The fence agent for Brocade FC switches.

Table B.4. Bull PAP (Platform Administration Processor)

Field	Description
Name	A name for the Bull PAP system connected to the cluster.
IP Address	The IP address assigned to the PAP console.
Login	The login name used to access the PAP console.
Password	The password used to authenticate the connection to the PAP console.
Password Script (optional)	The script that supplies a password for access to the fence device. Using this supersedes the **Password** parameter.
Domain	Domain of the Bull PAP system to power cycle.
fence_bullpap	The fence agent for Bull's NovaScale machines controlled by PAP management consoles.

Table B.5. Cisco MDS (Red Hat Enterprise Linux 5.4 and later)

Field	Description
Name	A name for the Cisco MDS 9000 series device with SNMP enabled.

Field	Description
IP address or hostname	The IP address or hostname assigned to the device.
UDP/TCP port (optional)	The UDP/TCP port to use for connection with the device; the default value is 161.
Login	The login name used to access the device.
Password	The password used to authenticate the connection to the device.
Password Script (optional)	The script that supplies a password for access to the fence device. Using this supersedes the **Password** parameter.
SNMP version	The SNMP version to use (1, 2c, 3).
SNMP community	The SNMP community string.
SNMP security level	The SNMP security level (noAuthNoPriv, authNoPriv, authPriv).
SNMP authentication protocol	The SNMP authentication protocol (MD5, SHA).
SNMP privacy protocol	The SNMP privacy protocol (DES, AES).
SNMP privacy protocol password	The SNMP privacy protocol password.
SNMP privacy protocol script	The script that supplies a password for SNMP privacy protocol. Using this supersedes the **SNMP privacy protocol password** parameter.
Power wait	Number of seconds to wait after issuing a power off or power on command.
Port	The port.
fence_cisco_mds	The fence agent for Cisco MDS.

Table B.6. Cisco UCS (Red Hat Enterprise Linux 5.6 and later)

Field	Description
Name	A name for the Cisco UCS device.
IP Address	The IP address or hostname assigned to the device.
IP port (optional)	The TCP port to use to connect to the device.
Login	The login name used to access the device.
Password	The password used to authenticate the connection to the device.
Password Script (optional)	The script that supplies a password for access to the fence device. Using this supersedes the **Password** parameter.
Use SSL connections	Use SSL connections to communicate with the device.
Sub-organization	Additional path needed to access suborganization.
Power wait	Number of seconds to wait after issuing a power off or power on command.
Port	Name of virtual machine.
fence_cisco_ucs	The fence agent for Cisco UCS.

Table B.7. Dell DRAC

Field	Description
Name	The name assigned to the DRAC.
IP Address	The IP address assigned to the DRAC.
Login	The login name used to access the DRAC.
Password	The password used to authenticate the connection to the DRAC.

Field	Description
Module Name	(optional) The module name for the DRAC when you have multiple DRAC modules.
Password Script (optional)	The script that supplies a password for access to the fence device. Using this supersedes the **Password** parameter.
Use SSH (DRAC5 only)	(Red Hat Enterprise Linux 5.4 and later) Indicates that system will use SSH to access the device.
Power wait	Number of seconds to wait after issuing a power off or power on command.
fence_drac	The fence agent for Dell Remote Access Card (DRAC).

Table B.8. Egenera SAN Controller

Field	Description
Name	A name for the BladeFrame device connected to the cluster.
CServer	The hostname (and optionally the username in the form of **username@hostname**) assigned to the device. Refer to the fence_egenera(8) man page for more information.
ESH Path (optional)	The path to the esh command on the cserver (default is /opt/pan- mgr/bin/esh)
lpan	The logical process area network (LPAN) of the device.
pserver	The processing blade (pserver) name of the device.
fence_egenera	The fence agent for the Egenera BladeFrame.

Table B.9. Fujitsu Siemens Remoteview Service Board (RSB)

Field	Description
Name	A name for the RSB to use as a fence device.
Hostname	The hostname assigned to the device.
Login	The login name used to access the device.
Password	The password used to authenticate the connection to the device.
Password Script (optional)	The script that supplies a password for access to the fence device. Using this supersedes the **Password** parameter.
fence_rsb	The fence agent for Fujitsu-Siemens RSB.

Table B.10. GNBD (Global Network Block Device)

Field	Description
Name	A name for the GNBD device used to fence the cluster. Note that the GFS server must be accessed via GNBD for cluster node fencing support.
Servers	The hostname of the server to fence the client from, in either IP address or hostname form. For multiple hostnames, separate each hostname with a whitespace.
IP Address	The cluster name of the node to be fenced. Refer to the **fence_gnbd**(8) man page for more information.
fence_gnbd	The fence agent for GNBD-based GFS clusters.

Table B.11. HP iLO (Integrated Lights Out)

Field	Description
Name	A name for the server with HP iLO support.
Hostname	The hostname assigned to the device.
Login	The login name used to access the device.
Password	The password used to authenticate the connection to the device.
Password Script (optional)	The script that supplies a password for access to the fence device. Using this supersedes the **Password** parameter.
Use SSL connections	Use SSL connections to communicate with the device.
Power wait	Number of seconds to wait after issuing a power off or power on command.
fence_ilo	The fence agent for HP servers with the Integrated Light Out (iLO) PCI card.

Table B.12. HP iLO (Integrated Lights Out) MP (Red Hat Enterprise Linux 5.5 and later)

Field	Description
Name	A name for the server with HP iLO support.
Hostname	The hostname assigned to the device.
IP port (optional)	TCP port to use for connection with the device.
Login	The login name used to access the device.
Password	The password used to authenticate the connection to the device.
Password Script (optional)	The script that supplies a password for access to the fence device. Using this supersedes the **Password** parameter.
Use SSH	(Red Hat Enterprise Linux 5.4 and later) Indicates that system will use SSH to access the device.
Path to SSH identity file	The identity file for SSH.
Force command prompt	The command prompt to use. The default value is 'MP>', 'hpiLO->'.
Power wait	Number of seconds to wait after issuing a power off or power on command.
fence_ilo_mp	The fence agent for HP iLO MP devices.

Table B.13. IBM Blade Center

Field	Description
Name	A name for the IBM BladeCenter device connected to the cluster.
IP Address	The IP address assigned to the device.
Login	The login name used to access the device.
Password	The password used to authenticate the connection to the device.
Password Script (optional)	The script that supplies a password for access to the fence device. Using this supersedes the **Password** parameter.
Power wait	Number of seconds to wait after issuing a power off or power on command.
Blade	The blade of the device.
Use SSH	(Red Hat Enterprise Linux 5.4 and later) Indicates that system will use SSH to access the device.
fence_bladecenter	The fence agent for IBM BladeCenter.

Table B.14. IBM iPDU (Red Hat Enterprise Linux 5.9 and later)

Field	Description
Name	A name for the IBM iPDU device connected to the cluster into which the fence daemon logs via the SNMP protocol.
IP Address	The IP address or hostname assigned to the device.
UDP/TCP Port	The UDP/TCP port to use for connection with the device; the default value is 161.
Login	The login name used to access the device.
Password	The password used to authenticate the connection to the device.
Password Script (optional)	The script that supplies a password for access to the fence device. Using this supersedes the **Password** parameter.
SNMP version	The SNMP version to use (1, 2c, 3); the default value is 1.
SNMP community	The SNMP community string; the default value is **private**.
SNMP security level	The SNMP security level (noAuthNoPriv, authNoPriv, authPriv).
SNMP authentication protocol	The SNMP authentication protocol (MD5, SHA).
SNMP privacy protocol	The SNMP privacy protocol (DES, AES).
SNMP privacy protocol password	The SNMP privacy protocol password.
SNMP privacy protocol script	The script that supplies a password for SNMP privacy protocol. Using this supersedes the **SNMP privacy protocol password** parameter.
Power wait	Number of seconds to wait after issuing a power off or power on command.
Port	The port.
fence_ipdu	The fence agent for iPDU over SNMP.

Table B.15. IBM Remote Supervisor Adapter II (RSA II)

Field	Description
Name	A name for the RSA device connected to the cluster.
Hostname	The hostname assigned to the device.
Login	The login name used to access the device.
Password	The password used to authenticate the connection to the device.
Password Script (optional)	The script that supplies a password for access to the fence device. Using this supersedes the **Password** parameter.
Power wait	Number of seconds to wait after issuing a power off or power on command.
fence_rsa	The fence agent for the IBM RSA II management interface.

Table B.16. IF MIB (Red Hat Enterprise Linux 5.6 and later)

Field	Description
Name	A name for the IF MIB device connected to the cluster.
IP address or hostname	The IP address or hostname assigned to the device.
UDP/TCP port (optional)	The UDP/TCP port to use for connection with the device; the default value is 161.
Login	The login name used to access the device.
Password	The password used to authenticate the connection to the device.

Field	Description
Password Script (optional)	The script that supplies a password for access to the fence device. Using this supersedes the **Password** parameter.
SNMP version	The SNMP version to use (1, 2c, 3); the default value is 1.
SNMP community	The SNMP community string.
SNMP security level	The SNMP security level (noAuthNoPriv, authNoPriv, authPriv).
SNMP authentication protocol	The SNMP authentication protocol (MD5, SHA).
SNMP privacy protocol	The SNMP privacy protocol (DES, AES).
SNMP privacy protocol password	The SNMP privacy protocol password.
SNMP privacy protocol script	The script that supplies a password for SNMP privacy protocol. Using this supersedes the **SNMP privacy protocol password** parameter.
Power wait	Number of seconds to wait after issuing a power off or power on command.
Port	Physical plug number or name of virtual machine.
fence_ifmib	The fence agent for IF-MIB devices.

Table B.17. IPMI (Intelligent Platform Management Interface) LAN

Field	Description
Name	A name for the IPMI LAN device connected to the cluster.
IP Address	The IP address assigned to the IPMI port.
Login	The login name of a user capable of issuing power on/off commands to the given IPMI port.
Password	The password used to authenticate the connection to the IPMI port.
Password Script (optional)	The script that supplies a password for access to the fence device. Using this supersedes the **Password** parameter.
Authentication Type	**none**, **password**, **md2**, or **md5**.
Privilege Level	The privilege level on the IPMI device.
Use Lanplus	**True** or **1**. If blank, then value is **False**.
fence_ipmilan	The fence agent for machines controlled by IPMI.

Table B.18. Manual Fencing

Field	Description
Name	A name to assign the Manual fencing agent. Refer to the **fence_manual**(8) man page for more information.

Warning

Manual fencing is *not* supported for production environments.

Table B.19. McData SAN Switch

Field	Description
Name	A name for the McData device connected to the cluster.

Field	Description
IP Address	The IP address assigned to the device.
Login	The login name used to access the device.
Password	The password used to authenticate the connection to the device.
Password Script (optional)	The script that supplies a password for access to the fence device. Using this supersedes the **Password** parameter.
Port	The switch outlet number.
fence_mcdata	The fence agent for McData FC switches.

Table B.20. QLogic SANBox2 Switch

Field	Description
Name	A name for the SANBox2 device connected to the cluster.
IP Address	The IP address assigned to the device.
Login	The login name used to access the device.
Password	The password used to authenticate the connection to the device.
Password Script (optional)	The script that supplies a password for access to the fence device. Using this supersedes the **Password** parameter.
Power wait	Number of seconds to wait after issuing a power off or power on command.
Port	The switch outlet number.
fence_sanbox2	The fence agent for QLogic SANBox2 FC switches.

Table B.21. RHEV-M REST API (RHEL 5.8 and later against RHEV 3.0 and later)

Field	Description
Name	Name of the RHEV-M REST API fencing device.
IP Address	The IP address or hostname assigned to the device.
IP port (optional)	The TCP port to use for connection with the device.
Login	The login name used to access the device.
Password	The password used to authenticate the connection to the device.
Password Script (optional)	The script that supplies a password for access to the fence device. Using this supersedes the **Password** parameter.
Use SSL connections	Use SSL connections to communicate with the device.
Power wait	Number of seconds to wait after issuing a power off or power on command.
Port	Physical plug number or name of virtual machine.
fence_rhevm	The fence agent for RHEV-M REST API.

Table B.22. RPS-10 Power Switch (two-node clusters only)

Field	Description
Name	A name for the WTI RPS-10 power switch connected to the cluster.
Device Name	The device name of the device the switch is connected to on the controlling host (for example, **/dev/ttys2**).

Field	Description
Port	The switch outlet number.
fence_wti	The fence agent for the WTI Network Power Switch.

Table B.23. SCSI Fencing

Field	Description
Name	A name for the SCSI fence device.
Node name	Name of the node to be fenced. Refer to the **fence_scsi**(8) man page for more information.
fence_scsi	The fence agent for SCSI persistent reservations.

Note

Use of SCSI persistent reservations as a fence method is supported with the following limitations:

- As of Red Hat Enterprise Linux 5.5 and fully-updated releases of Red Hat Enterprise Linux 5.4, SCSI fencing can be used in a 2-node cluster; previous releases did not support this feature.
- When using SCSI fencing, all nodes in the cluster must register with the same devices so that each node can remove another node's registration key from all the devices it is registered with.
- Devices used for the cluster volumes should be a complete LUN, not partitions. SCSI persistent reservations work on an entire LUN, meaning that access is controlled to each LUN, not individual partitions.
- As of Red Hat Enterprise Linux 5.5 and fully-updated releases of Red Hat Enterprise Linux 5.4, SCSI fencing can be used in conjunction with qdisk; previous releases did not support this feature. You cannot use **fence_scsi** on the LUN where **qdiskd** resides; it must be a raw LUN or raw partition of a LUN.

Table B.24. Virtual Machine Fencing

Field	Description
Name	Name of the virtual machine fencing device.
Domain	Unique domain name of the guest to be fenced.

Table B.25. VMware (SOAP Interface) (Red Hat Enterprise Linux 5.7 and later)

Field	Description
Name	Name of the virtual machine fencing device.
Hostname	The IP address or hostname assigned to the device.
IP port (optional)	The TCP port to use for connection with the device.
Login	The login name used to access the device.
Password	The password used to authenticate the connection to the device.
Password Script (optional)	The script that supplies a password for access to the fence device. Using this supersedes the **Password** parameter.

Field	Description
Use SSL connections	Use SSL connections to communicate with the device.
Power wait	Number of seconds to wait after issuing a power off or power on command.
Virtual machine name	Name of virtual machine in inventory path format (e.g., /datacenter/vm/Discovered_virtual_machine/myMachine).
Virtual machine UUID	The UUID of the virtual machine to fence.
fence_vmware_soap	The fence agent for VMWare over SOAP API.

Table B.26. Vixel SAN Switch

Field	Description
Name	A name for the Vixel switch connected to the cluster.
IP Address	The IP address assigned to the device.
Password	The password used to authenticate the connection to the device.
Password Script (optional)	The script that supplies a password for access to the fence device. Using this supersedes the **Password** parameter.
Port	The switch outlet number.
fence_vixel	The fence agent for Vixel switches.

Table B.27. WTI Power Switch

Field	Description
Name	A name for the WTI power switch connected to the cluster.
IP Address	The IP address assigned to the device.
Password	The password used to authenticate the connection to the device.
Password Script (optional)	The script that supplies a password for access to the fence device. Using this supersedes the **Password** parameter.
Power wait	Number of seconds to wait after issuing a power off or power on command.
Port	The switch outlet number.
Use SSH	(Red Hat Enterprise Linux 5.4 and later) Indicates that system will use SSH to access the device.
fence_wti	The fence agent for the WTI network power switch.

HA Resource Parameters

This appendix provides descriptions of HA resource parameters. You can configure the parameters with **Luci**, **system-config-cluster**, or by editing **etc/cluster/cluster.conf**. Table C.1, "HA Resource Summary" lists the resources, their corresponding resource agents, and references to other tables containing parameter descriptions. To understand resource agents in more detail you can view them in **/usr/share/cluster** of any cluster node.

For a comprehensive list and description of **cluster.conf** elements and attributes, refer to the cluster schema at **/usr/share/system-config-cluster/misc/cluster.ng**, and the annotated schema at **/usr/share/doc/system-config-cluster-X.Y.ZZ/cluster_conf.html** (for example **/usr/share/doc/system-config-cluster-1.0.57/cluster_conf.html**).

Table C.1. HA Resource Summary

Resource	Resource Agent	Reference to Parameter Description
Apache	apache.sh	Table C.2, "Apache Server"
File System	fs.sh	Table C.3, "File System"
GFS File System	clusterfs.sh	Table C.4, "GFS"
IP Address	ip.sh	Table C.5, "IP Address"
LVM	lvm.sh	Table C.6, "LVM"
MySQL	mysql.sh	Table C.7, "MySQL"
NFS Client	nfsclient.sh	Table C.8, "NFS Client"
NFS Export	nfsexport.sh	Table C.9, "NFS Export"
NFS Mount	netfs.sh	Table C.10, "NFS Mount"
Open LDAP	openldap.sh	Table C.11, "Open LDAP"
Oracle 10g Failover Instance	oracledb.sh	Table C.12, "Oracle 10g Failover Instance"
Oracle DB Agent	orainstance.sh	Table C.13, "Oracle DB"
Oracle Listener Agent	oralistener.sh	Table C.14, "Oracle Listener Agent"
PostgreSQL 8	postgres-8.sh	Table C.15, "PostgreSQL 8"
SAP Database	SAPDatabase	Table C.16, "SAP Database"
SAP Instance	SAPInstance	Table C.17, "SAP Instance"
Samba	smb.sh	Table C.18, "Samba Service"
Script	script.sh	Table C.19, "Script"
Service	service.sh	Table C.20, "Service"
Sybase ASE	ASEHAagent.sh	Table C.21, "Sybase ASE Failover Instance"
Tomcat 5	tomcat-5.sh	Table C.22, "Tomcat 5"
Virtual Machine	vm.sh	Table C.23, "Virtual Machine" NOTE: **Luci** displays this as a virtual service if the host cluster can support virtual machines.

Table C.2. Apache Server

Field	Description
Name	The name of the Apache Service.
Server Root	The default value is **/etc/httpd**.
Config File	Specifies the Apache configuration file. The default valuer is **/etc/httpd/conf**.
httpd Options	Other command line options for **httpd**.
Shutdown Wait (seconds)	Specifies the number of seconds to wait for correct end of service shutdown.

Table C.3. File System

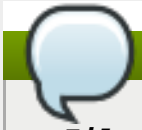
Field	Description
Name	Specifies a name for the file system resource.
File system type	If not specified, **mount** tries to determine the file system type.
Mount point	Path in file system hierarchy to mount this file system.
Device	Specifies the device associated with the file system resource. This can be a block device, file system label, or UUID of a file system.
Options	Mount options; that is, options used when the file system is mounted. These may be file-system specific. Refer to the ***mount***(8) man page for supported mount options.
File system ID	 **Note** ***File System ID*** is used only by NFS services. When creating a new file system resource, you can leave this field blank. Leaving the field blank causes a file system ID to be assigned automatically after you commit the parameter during configuration. If you need to assign a file system ID explicitly, specify it in this field.
Force unmount	If enabled, forces the file system to unmount. The default setting is ***disabled***. ***Force Unmount*** kills all processes using the mount point to free up the mount when it tries to unmount.
Reboot host node if unmount fails	If enabled, reboots the node if unmounting this file system fails. The default setting is ***disabled***.
Check file system before mounting	If enabled, causes **fsck** to be run on the file system before mounting it. The default setting is ***disabled***.
Enable NFS daemon and lockd workaround	If your filesystem is exported via NFS and occasionally fails to unmount (either during shutdown or service relocation), setting this option will drop all filesystem references prior to the unmount operation. Setting this option requires that you enable the **Force unmount** option. You should set this option as a last resort only, as this is a hard attempt to unmount a file system. You can enable NFS lock workarounds in a soft attempt to unmount a file system at the level of cluster service configuration, as described in Section 3.9, "Adding a Cluster Service to the Cluster".

Table C.4. GFS

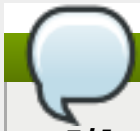
Field	Description
Name	The name of the file system resource.
Mount point	The path to which the file system resource is mounted.
Device	The device file associated with the file system resource.
File system type	Specify GFS or GFS2.
Options	Mount options.
File system ID	 **Note** ***File System ID*** is used only by NFS services. When creating a new GFS resource, you can leave this field blank. Leaving the field blank causes a file system ID to be assigned automatically after you commit the parameter during configuration. If you need to assign a file system ID explicitly, specify it in this field.
Force unmount	If enabled, forces the file system to unmount. The default setting is ***disabled***. ***Force Unmount*** kills all processes using the mount point to free up the mount when it tries to unmount. With GFS resources, the mount point is *not* unmounted at service tear-down unless ***Force Unmount*** is *enabled*.
Reboot host node if unmount fails	If enabled and unmounting the file system fails, the node will immediately reboot. Generally, this is used in conjunction with force-unmount support, but it is not required.
Enable NFS daemon and lockd workaround	If your filesystem is exported via NFS and occasionally fails to unmount (either during shutdown or service relocation), setting this option will drop all filesystem references prior to the unmount operation. Setting this option requires that you enable the **Force unmount** option. You should set this option as a last resort only, as this is a hard attempt to unmount a file system. You can enable NFS lock workarounds in a soft attempt to unmount a file system at the level of cluster service configuration, as described in Section 3.9, “Adding a Cluster Service to the Cluster”.

Table C.5. IP Address

Field	Description
IP address	The IP address for the resource. This is a virtual IP address. IPv4 and IPv6 addresses are supported, as is NIC link monitoring for each IP address.
Monitor link	Enabling this causes the status check to fail if the link on the NIC to which this IP address is bound is not present.

Table C.6. LVM

Field	Description
Name	A unique name for this LVM resource.
Volume Group Name	A descriptive name of the volume group being managed.
Logical Volume Name	Name of the logical volume being managed. This parameter is optional if there is more than one logical volume in the volume group being managed.

Field	Description
Fence the node if it is unable to clean up LVM tags	Fence the node if it is unable to clean up LVM tags.

Table C.7. MySQL

Field	Description
Name	Specifies a name of the MySQL server resource.
Config File	Specifies the configuration file. The default value is **/etc/my.cnf**.
Listen Address	Specifies an IP address for MySQL server. If an IP address is not provided, the first IP address from the service is taken.
mysqld Options	Other command line options for **httpd**.
Shutdown Wait (seconds)	Specifies the number of seconds to wait for correct end of service shutdown.

Table C.8. NFS Client

Field	Description
Name	This is a symbolic name of a client used to reference it in the resource tree. This is *not* the same thing as the ***Target*** option.
Target	This is the server from which you are mounting. It can be specified using a hostname, a wildcard (IP address or hostname based), or a netgroup defining a host or hosts to export to.
Options	Defines a list of options for this client — for example, additional client access rights. For more information, refer to the ***exports*** *(5)* man page, *General Options*.
Allow Recover	Allow recovery of the NFS client.

Table C.9. NFS Export

Field	Description
Name	Descriptive name of the resource. The NFS Export resource ensures that NFS daemons are running. It is fully reusable; typically, only one NFS Export resource is needed. **Tip** Name the NFS Export resource so it is clearly distinguished from other NFS resources.

Table C.10. NFS Mount

Field	Description

Field	Description
Name	Symbolic name for the NFS mount. **Note** This resource is required only when a cluster service is configured to be an NFS client.
Mount point	Path to which the file system resource is mounted.
Host	NFS server IP address or hostname.
Export path	NFS Export directory name.
NFS version	NFS protocol: - ***NFS3*** — Specifies using NFSv3 protocol. The default setting is ***NFS3***. - ***NFS4*** — Specifies using NFSv4 protocol.
Options	Mount options. Specifies a list of mount options. If none are specified, the NFS file system is mounted **`-o sync`**. For more information, refer to the ***nfs****(5)* man page.
Force unmount	If ***`Force unmount`*** is enabled, the cluster kills all processes using this file system when the service is stopped. Killing all processes using the file system frees up the file system. Otherwise, the unmount will fail, and the service will be restarted.

Table C.11. Open LDAP

Field	Description
Name	Specifies a service name for logging and other purposes.
Config File	Specifies an absolute path to a configuration file. The default value is **`/etc/openldap/slapd.conf`**.
URL List	The default value is **`ldap:///`**.
`slapd` Options	Other command line options for **`slapd`**.
Shutdown Wait (seconds)	Specifies the number of seconds to wait for correct end of service shutdown.

Table C.12. Oracle 10g Failover Instance

Field	Description
Instance name (SID) of Oracle instance	Instance name.
Oracle user name	This is the user name of the Oracle user that the Oracle AS instance runs as.
Oracle application home directory	This is the Oracle (application, not user) home directory. It is configured when you install Oracle.
Virtual hostname (optional)	Virtual Hostname matching the installation hostname of Oracle 10g. Note that during the start/stop of an oracledb resource, your hostname is changed temporarily to this hostname. Therefore, you should configure an oracledb resource as part of an exclusive service only.

Table C.13. Oracle DB

Field	Description
Instance name (SID) of Oracle instance	Instance name.
Oracle user name	This is the user name of the Oracle user that the Oracle instance runs as.
Oracle application home directory	This is the Oracle (application, not user) home directory. It is configured when you install Oracle.
List of Oracle listeners (optional, separated by spaces)	List of Oracle listeners which will be started with the database instance. Listener names are separated by whitespace. Defaults to empty which disables listeners.
Path to lock file (optional)	Location for lockfile which will be used for checking if the Oracle should be running or not. Defaults to location under **/tmp**.

Table C.14. Oracle Listener Agent

Field	Description
Listener Name	Listener name.
Oracle user name	This is the user name of the Oracle user that the Oracle instance runs as.
Oracle application home directory	This is the Oracle (application, not user) home directory. It is configured when you install Oracle.

Table C.15. PostgreSQL 8

Field	Description
Name	Specifies a service name for logging and other purposes.
Config File	Define absolute path to configuration file. The default value is **/var/lib/pgsql/data/postgresql.conf**.
Postmaster User	User who runs the database server because it cannot be run by root. The default value is postgres.
Postmaster Options	Other command line options for postmaster.
Startup Wait (seconds)	Specifies the number of seconds to wait for correct end of service startup.
Shutdown Wait (seconds)	Specifies the number of seconds to wait for correct end of service shutdown.

Table C.16. SAP Database

Field	Description
SAP Database Name	Specifies a unique SAP system identifier. For example, P01.
SAP executable directory	Specifies the fully qualified path to **sapstartsrv** and **sapcontrol**.

Field	Description
Database type	Specifies one of the following database types: Oracle, DB6, or ADA.
Oracle TNS listener name	Specifies Oracle TNS listener name.
ABAP stack is not installed, only Java stack is installed	If you do not have an ABAP stack installed in the SAP database, enable this parameter.
Application Level Monitoring	Activates application level monitoring.
Automatic Startup Recovery	Enable or disable automatic startup recovery.
Path to Java SDK	Path to Java SDK.
File name of the JDBC Driver	File name of the JDBC driver.
Path to a pre-start script	Path to a pre-start script.
Path to a post-start script	Path to a post-start script.
Path to a pre-stop script	Path to a pre-stop script
Path to a post-stop script	Path to a post-stop script
J2EE instance bootstrap directory	The fully qualified path the J2EE instance bootstrap directory. For example, **/usr/sap/P01/J00/j2ee/cluster/bootstrap**.
J2EE security store path	The fully qualified path the J2EE security store directory. For example, **/usr/sap/P01/SYS/global/security/lib/tools**.

Table C.17. SAP Instance

Field	Description
SAP Instance Name	The fully qualified SAP instance name. For example, P01_DVEBMGS00_sapp01ci.
SAP executable directory	The fully qualified path to **sapstartsrv** and **sapcontrol**.
Directory containing the SAP START profile	The fully qualified path to the SAP START profile.
Name of the SAP START profile	Specifies name of the SAP START profile.

Field	Description
Number of seconds to wait before checking startup status	Specifies the number of seconds to wait before checking the startup status (do not wait for J2EE-Addin).
Enable automatic startup recovery	Enable or disable automatic startup recovery.
Path to a pre-start script	Path to a pre-start script.
Path to a post-start script	Path to a post-start script.
Path to a pre-stop script	Path to a pre-stop script
Path to a post-stop script	Path to a post-stop script

Note

Regarding Table C.18, "Samba Service", when creating or editing a cluster service, connect a Samba-service resource directly to the service, *not* to a resource within a service.

Note

Red Hat Enterprise Linux 5 does not support running Clustered Samba in an active/active configuration, in which Samba serves the same shared file system from multiple nodes. Red Hat Enterprise Linux 5 does support running Samba in a cluster in active/passive mode, with failover from one node to the other nodes in a cluster. Note that if failover occurs, locking states are lost and active connections to Samba are severed so that the clients must reconnect.

Table C.18. Samba Service

Field	Description
Name	Specifies the name of the Samba server.
Workgroup	Specifies a Windows workgroup name or Windows NT domain of the Samba service.

Table C.19. Script

Field	Description
Name	Specifies a name for the custom user script. The script resource allows a standard LSB-compliant init script to be used to start a clustered service.
Full path to script file	Enter the path where this custom script is located (for example, **/etc/init.d/*userscript***).

Table C.20. Service

Field	Description
Service name	Name of service. This defines a collection of resources, known as a resource group or cluster service.
Automatically start this service	If enabled, this service (or resource group) is started automatically after the cluster forms a quorum. If this parameter is *disabled*, this service is *not* started automatically after the cluster forms a quorum; the service is put into the ***disabled*** state.
Run exclusive	If enabled, this service (resource group) can only be relocated to run on another node exclusively; that is, to run on a node that has no other services running on it. If no nodes are available for a service to run exclusively, the service is not restarted after a failure. Additionally, other services do not automatically relocate to a node running this service as ***Run exclusive***. You can override this option by manual start or relocate operations.
Failover Domain	Defines lists of cluster members to try in the event that a service fails. For information on configuring a failover domain with Conga, refer to Section 3.7, "Configuring a Failover Domain". For information on configuring a failover domain with **system-config-cluster**, refer to Section 5.6, "Configuring a Failover Domain".
Recovery policy	***Recovery policy*** provides the following options: - ***Disable*** — Disables the resource group if any component fails. - ***Relocate*** — Tries to restart service in another node; that is, it does not try to restart in the current node. - ***Restart*** — Tries to restart failed parts of this service locally (in the current node) before trying to relocate (default) to service to another node. - ***Restart-Disable*** — (Red Hat Enterprise Linux release 5.6 and later) The service will be restarted in place if it fails. However, if restarting the service fails the service will be disabled instead of being moved to another host in the cluster.

Table C.21. Sybase ASE Failover Instance

Field	Description
Instance Name	Specifies the instance name of the Sybase ASE resource.
ASE server name	The ASE server name that is configured for the HA service.
SYBASE home directory	The home directory of Sybase products.
Login file	The full path of login file that contains the login-password pair.
Interfaces file	The full path of the interfaces file that is used to start/access the ASE server.
SYBASE_ASE directory name	The directory name under sybase_home where ASE products are installed.
SYBASE_OCS directory name	The directory name under sybase_home where OCS products are installed. For example, ASE-15_0.
Sybase user	The user who can run ASE server.
Deep probe timeout	The maximum seconds to wait for the response of ASE server before determining that the server had no response while running deep probe.

Table C.22. Tomcat 5

Field	Description
Name	Specifies a service name for logging and other purposes.
Config File	Specifies the absolute path to the configuration file. The default value is **/etc/tomcat5/tomcat5.conf**.
Tomcat User	User who runs the Tomcat server. The default value is *tomcat*.
Catalina Options	Other command line options for Catalina.
Catalina Base	Catalina base directory (differs for each service) The default value is /usr/share/tomcat5.
Shutdown Wait (seconds)	Specifies the number of seconds to wait for correct end of service shutdown. The default value is 30.

Important

Regarding Table C.23, "Virtual Machine", when you configure your cluster with virtual machine resources, you should use the **rgmanager** tools to start and stop the virtual machines. Using **xm** or **virsh** to start the machine can result in the virtual machine running in more than one place, which can cause data corruption in the virtual machine. For information on configuring your system to reduce the chances of administrators accidentally "double-starting" virtual machines by using both cluster and non-cluster tools, refer to Section 2.12, "Configuring Virtual Machines in a Clustered Environment".

Note

Virtual machine resources are configured differently than other cluster resources; they are configured as services. To configure a virtual machine resource with **luci**, at the detailed menu for the cluster (below the **clusters** menu), click **Services**, then click **Add a Virtual Machine Service**. You can then enter the virtual machine resource parameters. For information on configuring cluster services, refer to Section 3.9, "Adding a Cluster Service to the Cluster".

Table C.23. Virtual Machine

Field	Description
Virtual machine name	Specifies the name of the virtual machine.
Path to VM configuration files	A colon-delimited path specification that **xm create** searches for the virtual machine configuration file. For example: **/etc/xen:/guests/config_files:/var/xen/configs** **Important** The path should *never* directly point to a virtual machine configuration file.

Field	Description
VM Migration Mapping	Specifies an alternate interface for migration. You can specify this when, for example, the network address used for virtual machine migration on a node differs from the address of the node used for cluster communication. Specifying the following indicates that when you migrate a virtual machine from **member** to **member2**, you actually migrate to **target2**. Similarly, when you migrate from **member2** to **member**, you migrate using **target**. **member:target,member2:target2**
Migration type	Specifies a migration type of ***live*** or ***pause***. The default setting is ***live***.
Hypervisor	Hypervisor URI (automatic, KVM, or Xen)
Automatically start this service	If enabled, this virtual machine is started automatically after the cluster forms a quorum. If this parameter is *disabled*, this virtual machine service is *not* started automatically after the cluster forms a quorum; the virtual machine is put into the ***disabled*** state.
Run exclusive	If enabled, this virtual machine can only be relocated to run on another node exclusively; that is, to run on a node that has no other virtual machines running on it. If no nodes are available for a virtual machine to run exclusively, the virtual machine is not restarted after a failure. Additionally, other virtual machines do not automatically relocate to a node running this virtual machine as ***Run exclusive***. You can override this option by manual start or relocate operations.
Failover Domain	Defines lists of cluster members to try in the event that a virtual machine fails.
Recovery policy	***Recovery policy*** provides the following options: - ***Disable*** — Disables the virtual machine if it fails. - ***Relocate*** — Tries to restart the virtual machine in another node; that is, it does not try to restart in the current node. - ***Restart*** — Tries to restart the virtual machine locally (in the current node) before trying to relocate (default) to virtual machine to another node. - ***Restart-Disable*** — (Red Hat Enterprise Linux Release 5.6 and later) The service will be restarted in place if it fails. However, if restarting the service fails the service will be disabled instead of being moved to another host in the cluster.
Maximum number of restart failures before relocating	Maximum number of restarts for an independent subtree before giving up.
Length of time in seconds after which to forget a restart	Amount of time before a failure is forgotten for an independent subtree.

HA Resource Behavior

This appendix describes common behavior of HA resources. It is meant to provide ancillary information that may be helpful in configuring HA services. You can configure the parameters with **Luci**, **system-config-cluster**, or by editing **etc/cluster/cluster.conf**. For descriptions of HA resource parameters, refer to Appendix C, *HA Resource Parameters*. To understand resource agents in more detail you can view them in **/usr/share/cluster** of any cluster node.

Note

To fully comprehend the information in this appendix, you may require detailed understanding of resource agents and the cluster configuration file, **/etc/cluster/cluster.conf**.

An HA service is a group of cluster resources configured into a coherent entity that provides specialized services to clients. An HA service is represented as a resource tree in the cluster configuration file, **/etc/cluster/cluster.conf** (in each cluster node). In the cluster configuration file, each resource tree is an XML representation that specifies each resource, its attributes, and its relationship among other resources in the resource tree (parent, child, and sibling relationships).

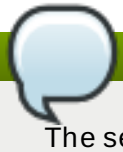
Note

Because an HA service consists of resources organized into a hierarchical tree, a service is sometimes referred to as a *resource tree* or *resource group*. Both phrases are synonymous with *HA service*.

At the root of each resource tree is a special type of resource — a *service resource*. Other types of resources comprise the rest of a service, determining its characteristics. Configuring an HA service consists of creating a service resource, creating subordinate cluster resources, and organizing them into a coherent entity that conforms to hierarchical restrictions of the service.

This appendix consists of the following sections:

- Section D.1, “Parent, Child, and Sibling Relationships Among Resources”
- Section D.2, “Sibling Start Ordering and Resource Child Ordering”
- Section D.3, “Inheritance, the <resources> Block, and Reusing Resources”
- Section D.4, “Failure Recovery and Independent Subtrees”
- Section D.5, “Debugging and Testing Services and Resource Ordering”



Note

The sections that follow present examples from the cluster configuration file, **/etc/cluster/cluster.conf**, for illustration purposes only.

D.1. Parent, Child, and Sibling Relationships Among Resources

A cluster service is an integrated entity that runs under the control of **rgmanager**. All resources in a service run on the same node. From the perspective of **rgmanager**, a cluster service is one entity that can be started, stopped, or relocated. Within a cluster service, however, the hierarchy of the resources determines the order in which each resource is started and stopped.The hierarchical levels consist of parent, child, and sibling.

Example D.1, “Resource Hierarchy of Service foo” shows a sample resource tree of the service *foo*. In the example, the relationships among the resources are as follows:

- **fs:myfs** (<fs name="myfs" ...>) and **ip:10.1.1.2** (<ip address="10.1.1.2 .../>) are siblings.
- **fs:myfs** (<fs name="myfs" ...>) is the parent of **script:script_child** (<script name="script_child"/>).
- **script:script_child** (<script name="script_child"/>) is the child of **fs:myfs** (<fs name="myfs" ...>).

Example D.1. Resource Hierarchy of Service foo

```
<service name="foo" ...>
    <fs name="myfs" ...>
        <script name="script_child"/>
    </fs>
    <ip address="10.1.1.2" .../>
</service>
```

The following rules apply to parent/child relationships in a resource tree:

- Parents are started before children.
- Children must all stop cleanly before a parent may be stopped.
- For a resource to be considered in good health, all its children must be in good health.

D.2. Sibling Start Ordering and Resource Child Ordering

The Service resource determines the start order and the stop order of a child resource according to whether it designates a child-type attribute for a child resource as follows:

- Designates child-type attribute (*typed* child resource) — If the Service resource designates a child-type attribute for a child resource, the child resource is *typed*. The child-type attribute explicitly determines the start and the stop order of the child resource.
- *Does not designate* child-type attribute (*non-typed* child resource) — If the Service resource *does not designate* a child-type attribute for a child resource, the child resource is *non-typed*. The Service resource does not explicitly control the starting order and stopping order of a non-typed child resource. However, a non-typed child resource is started and stopped according to its order in **/etc/cluster/cluster.conf** In addition, non-typed child resources are started after all typed child resources have started and are stopped before any typed child resources have stopped.

Note

The only resource to implement defined *child resource type* ordering is the Service resource.

For more information about typed child resource start and stop ordering, refer to Section D.2.1, “Typed Child Resource Start and Stop Ordering”. For more information about non-typed child resource start and stop ordering, refer to Section D.2.2, “Non-typed Child Resource Start and Stop Ordering”.

D.2.1. Typed Child Resource Start and Stop Ordering

For a typed child resource, the type attribute for the child resource defines the start order and the stop order of each resource type with a number ranging from 1 to 100; one value for start, and one value for stop. The lower the number, the earlier a resource type starts or stops. For example, Table D.1, “Child Resource Type Start and Stop Order” shows the start and stop values for each resource type; Example D.2, “Resource Start and Stop Values: Excerpt from Service Resource Agent, **service.sh**” shows the start and stop values as they appear in the Service resource agent, **service.sh**. For the Service resource, all LVM children are started first, followed by all File System children, followed by all Script children, and so forth.

Table D.1. Child Resource Type Start and Stop Order

Resource	Child Type	Start-order Value	Stop-order Value
LVM	lvm	1	9
File System	fs	2	8
GFS File System	clusterfs	3	7
NFS Mount	netfs	4	6
NFS Export	nfsexport	5	5
NFS Client	nfsclient	6	4
IP Address	ip	7	2
Samba	smb	8	3
Script	script	9	1

Example D.2. Resource Start and Stop Values: Excerpt from Service Resource Agent, service.sh

```
<special tag="rgmanager">
    <attributes root="1" maxinstances="1"/>
    <child type="lvm" start="1" stop="9"/>
    <child type="fs" start="2" stop="8"/>
    <child type="clusterfs" start="3" stop="7"/>
    <child type="netfs" start="4" stop="6"/>
    <child type="nfsexport" start="5" stop="5"/>
    <child type="nfsclient" start="6" stop="4"/>
    <child type="ip" start="7" stop="2"/>
    <child type="smb" start="8" stop="3"/>
    <child type="script" start="9" stop="1"/>
</special>
```

Ordering within a resource type is preserved as it exists in the cluster configuration file, **/etc/cluster/cluster.conf**. For example, consider the starting order and stopping order of the typed child resources in Example D.3, “Ordering Within a Resource Type”.

Example D.3. Ordering Within a Resource Type

```
<service name="foo">
  <script name="1" .../>
```

```
  <lvm name="1" .../>
  <ip address="10.1.1.1" .../>
  <fs name="1" .../>
  <lvm name="2" .../>
</service>
```

Typed Child Resource Starting Order

In Example D.3, "Ordering Within a Resource Type", the resources are started in the following order:

1. **lvm:1** — This is an LVM resource. All LVM resources are started first. **lvm:1** (**<lvm name="1" .../>**) is the first LVM resource started among LVM resources because it is the first LVM resource listed in the Service *foo* portion of **/etc/cluster/cluster.conf**.

2. **lvm:2** — This is an LVM resource. All LVM resources are started first. **lvm:2** (**<lvm name="2" .../>**) is started after **lvm:1** because it is listed after **lvm:1** in the Service *foo* portion of **/etc/cluster/cluster.conf**.

3. **fs:1** — This is a File System resource. If there were other File System resources in Service *foo*, they would start in the order listed in the Service *foo* portion of **/etc/cluster/cluster.conf**.

4. **ip:10.1.1.1** — This is an IP Address resource. If there were other IP Address resources in Service *foo*, they would start in the order listed in the Service *foo* portion of **/etc/cluster/cluster.conf**.

5. **script:1** — This is a Script resource. If there were other Script resources in Service *foo*, they would start in the order listed in the Service *foo* portion of **/etc/cluster/cluster.conf**.

Typed Child Resource Stopping Order

In Example D.3, "Ordering Within a Resource Type", the resources are stopped in the following order:

1. **script:1** — This is a Script resource. If there were other Script resources in Service *foo*, they would stop in the reverse order listed in the Service *foo* portion of **/etc/cluster/cluster.conf**.

2. **ip:10.1.1.1** — This is an IP Address resource. If there were other IP Address resources in Service *foo*, they would stop in the reverse order listed in the Service *foo* portion of **/etc/cluster/cluster.conf**.

3. **fs:1** — This is a File System resource. If there were other File System resources in Service *foo*, they would stop in the reverse order listed in the Service *foo* portion of **/etc/cluster/cluster.conf**.

4. **lvm:2** — This is an LVM resource. All LVM resources are stopped last. **lvm:2** (**<lvm name="2" .../>**) is stopped before **lvm:1**; resources within a group of a resource type are stopped in the reverse order listed in the Service *foo* portion of **/etc/cluster/cluster.conf**.

5. **lvm:1** — This is an LVM resource. All LVM resources are stopped last. **lvm:1** (**<lvm name="1" .../>**) is stopped after **lvm:2**; resources within a group of a resource type are stopped in the reverse order listed in the Service *foo* portion of **/etc/cluster/cluster.conf**.

D.2.2. Non-typed Child Resource Start and Stop Ordering

Additional considerations are required for non-typed child resources. For a non-typed child resource, starting order and stopping order are not explicitly specified by the Service resource. Instead, starting order and stopping order are determined according to the order of the child resource in **/etc/cluster/cluster.conf**. Additionally, non-typed child resources are started after all typed child resources and stopped before any typed child resources.

For example, consider the starting order and stopping order of the non-typed child resources in Example D.4, "Non-typed and Typed Child Resource in a Service".

Example D.4. Non-typed and Typed Child Resource in a Service

```
<service name="foo">
  <script name="1" .../>
  <nontypedresource name="foo"/>
  <lvm name="1" .../>
  <nontypedresourcetwo name="bar"/>
  <ip address="10.1.1.1" .../>
  <fs name="1" .../>
  <lvm name="2" .../>
</service>
```

Non-typed Child Resource Starting Order

In Example D.4, "Non-typed and Typed Child Resource in a Service", the child resources are started in the following order:

1. **lvm:1** — This is an LVM resource. All LVM resources are started first. **lvm:1** (**<lvm name="1" .../>**) is the first LVM resource started among LVM resources because it is the first LVM resource listed in the Service *foo* portion of **/etc/cluster/cluster.conf**.

2. **lvm:2** — This is an LVM resource. All LVM resources are started first. **lvm:2** (**<lvm name="2" .../>**) is started after **lvm:1** because it is listed after **lvm:1** in the Service *foo* portion of **/etc/cluster/cluster.conf**.

3. **fs:1** — This is a File System resource. If there were other File System resources in Service *foo*, they would start in the order listed in the Service *foo* portion of **/etc/cluster/cluster.conf**.

4. **ip:10.1.1.1** — This is an IP Address resource. If there were other IP Address resources in Service *foo*, they would start in the order listed in the Service *foo* portion of **/etc/cluster/cluster.conf**.

5. **script:1** — This is a Script resource. If there were other Script resources in Service *foo*, they would start in the order listed in the Service *foo* portion of **/etc/cluster/cluster.conf**.

6. **nontypedresource:foo** — This is a non-typed resource. Because it is a non-typed resource, it is started after the typed resources start. In addition, its order in the Service resource is before the other non-typed resource, **nontypedresourcetwo:bar**; therefore, it is started before **nontypedresourcetwo:bar**. (Non-typed resources are started in the order that they appear in the Service resource.)

7. **nontypedresourcetwo:bar** — This is a non-typed resource. Because it is a non-typed resource, it is started after the typed resources start. In addition, its order in the Service resource is after the other non-typed resource, **nontypedresource:foo**; therefore, it is

started after **nontypedresource:foo**. (Non-typed resources are started in the order that they appear in the Service resource.)

Non-typed Child Resource Stopping Order

In Example D.4, "Non-typed and Typed Child Resource in a Service", the child resources are stopped in the following order:

1. **nontypedresourcetwo:bar** — This is a non-typed resource. Because it is a non-typed resource, it is stopped before the typed resources are stopped. In addition, its order in the Service resource is after the other non-typed resource, **nontypedresource:foo**; therefore, it is stopped before **nontypedresource:foo**. (Non-typed resources are stopped in the reverse order that they appear in the Service resource.)

2. **nontypedresource:foo** — This is a non-typed resource. Because it is a non-typed resource, it is stopped before the typed resources are stopped. In addition, its order in the Service resource is before the other non-typed resource, **nontypedresourcetwo:bar**; therefore, it is stopped after **nontypedresourcetwo:bar**. (Non-typed resources are stopped in the reverse order that they appear in the Service resource.)

3. **script:1** — This is a Script resource. If there were other Script resources in Service *foo*, they would stop in the reverse order listed in the Service *foo* portion of **/etc/cluster/cluster.conf**.

4. **ip:10.1.1.1** — This is an IP Address resource. If there were other IP Address resources in Service *foo*, they would stop in the reverse order listed in the Service *foo* portion of **/etc/cluster/cluster.conf**.

5. **fs:1** — This is a File System resource. If there were other File System resources in Service *foo*, they would stop in the reverse order listed in the Service *foo* portion of **/etc/cluster/cluster.conf**.

6. **lvm:2** — This is an LVM resource. All LVM resources are stopped last. **lvm:2** (**<lvm name="2" .../>**) is stopped before **lvm:1**; resources within a group of a resource type are stopped in the reverse order listed in the Service *foo* portion of **/etc/cluster/cluster.conf**.

7. **lvm:1** — This is an LVM resource. All LVM resources are stopped last. **lvm:1** (**<lvm name="1" .../>**) is stopped after **lvm:2**; resources within a group of a resource type are stopped in the reverse order listed in the Service *foo* portion of **/etc/cluster/cluster.conf**.

D.3. Inheritance, the <resources> Block, and Reusing Resources

Some resources benefit by inheriting values from a parent resource; that is commonly the case in an NFS service. Example D.5, "NFS Service Set Up for Resource Reuse and Inheritance" shows a typical NFS service configuration, set up for resource reuse and inheritance.

Example D.5. NFS Service Set Up for Resource Reuse and Inheritance

```
<resources>
  <nfsclient name="bob" target="bob.example.com" options="rw,no_root_squash"/>
  <nfsclient name="jim" target="jim.example.com" options="rw,no_root_squash"/>
  <nfsexport name="exports"/>
</resources>
```

```
<service name="foo">
  <fs name="1" mountpoint="/mnt/foo" device="/dev/sdb1" fsid="12344">
    <nfsexport ref="exports">  <!-- nfsexport's path and fsid
        attributes are inherited from the mountpoint and fsid
 attribute of the parent fs resource -->
    <nfsclient ref="bob"/> <!-- nfsclient's path is inherited
        from the mountpoint and the fsid is added to the options
 string during export -->
    <nfsclient ref="jim"/ >
  </nfsexport>
</fs>
<fs name="2" mountpoint="/mnt/bar" device="/dev/sdb2" fsid="12345">
  <nfsexport ref="exports">
    <nfsclient ref="bob"/> <!-- Because all of the critical
       data for this resource is either defined in the resources block
       or inherited, we can reference it again! -->
    <nfsclient ref="jim"/>
  </nfsexport>
</fs>
<ip address="10.2.13.20"/>
</service>
```

If the service were flat (that is, with no parent/child relationships), it would need to be configured as follows:

- The service would need four nfsclient resources — one per file system (a total of two for file systems), and one per target machine (a total of two for target machines).
- The service would need to specify export path and file system ID to each nfsclient, which introduces chances for errors in the configuration.

In Example D.5, “NFS Service Set Up for Resource Reuse and Inheritance” however, the NFS client resources *nfsclient:bob* and *nfsclient:jim* are defined once; likewise, the NFS export resource *nfsexport:exports* is defined once. All the attributes needed by the resources are inherited from parent resources. Because the inherited attributes are dynamic (and do not conflict with one another), it is possible to reuse those resources — which is why they are defined in the resources block. It may not be practical to configure some resources in multiple places. For example, configuring a file system resource in multiple places can result in mounting one file system on two nodes, therefore causing problems.

D.4. Failure Recovery and Independent Subtrees

In most enterprise environments, the normal course of action for failure recovery of a service is to restart the entire service if any component in the service fails. For example, in Example D.6, “Service *foo* Normal Failure Recovery”, if any of the scripts defined in this service fail, the normal course of action is to restart (or relocate or disable, according to the service recovery policy) the service. However, in some circumstances certain parts of a service may be considered non-critical; it may be necessary to restart only part of the service in place before attempting normal recovery. To accomplish that, you can use the **__independent_subtree** attribute. For example, in Example D.7, “Service *foo* Failure Recovery with **__independent_subtree** Attribute”, the **__independent_subtree** attribute is used to accomplish the following actions:

- If script:script_one fails, restart script:script_two and script:script_one.
- If script:script_two fails, restart just script:script_two.
- If script:script_three fails, restart script:script_one, script:script_two, and script:script_three.
- If script:script_four fails, restart the whole service.

Example D.6. Service *foo* Normal Failure Recovery

```
<service name="foo">
      <script name="script_one" ...>
          <script name="script_two" .../>
      </script>
      <script name="script_three" .../>
</service>
```

Example D.7. Service *foo* Failure Recovery with *__independent_subtree* Attribute

```
<service name="foo">
      <script name="script_one" __independent_subtree="1" ...>
          <script name="script_two" __independent_subtree="1" .../>
          <script name="script_three" .../>
      </script>
      <script name="script_four" .../>
</service>
```

In some circumstances, if a component of a service fails you may want to disable only that component without disabling the entire service, to avoid affecting other services the use other components of that service. As of the Red Hat Enterprise Linux 5.6 release, you can accomplish that by using the ***__independent_subtree="2"*** attribute, which designates the independent subtree as non-critical.

Note

You may only use the non-critical flag on singly-referenced resources. The non-critical flag works with all resources at all levels of the resource tree, but should not be used at the top level when defining services or virtual machines.

As of the Red Hat Enterprise Linux 5.6 release, you can set maximum restart and restart expirations on a per-node basis in the resource tree for independent subtrees. To set these thresholds, you can use the following attributes:

- **__max_restarts** configures the maximum number of tolerated restarts prior to giving up.
- **__restart_expire_time** configures the amount of time, in seconds, after which a restart is no longer attempted.

D.5. Debugging and Testing Services and Resource Ordering

You can debug and test services and resource ordering with the **rg_test** utility. **rg_test** is a command-line utility that is run from a shell or a terminal (it is not available in **Conga** or **system-config-cluster**.) Table D.2, "**rg_test** Utility Summary" summarizes the actions and syntax for the **rg_test** utility.

Table D.2. rg_test Utility Summary

Action	Syntax
Display the resource rules that **rg_test** understands.	**rg_test rules**
Test a configuration (and /usr/share/cluster) for errors or redundant resource agents.	**rg_test test /etc/cluster/cluster.conf**
Display the start and stop ordering of a service.	Display start order: **rg_test noop /etc/cluster/cluster.conf start service *servicename*** Display stop order: **rg_test noop /etc/cluster/cluster.conf stop service *servicename***
Explicitly start or stop a service.	 **Important** Only do this on one node, and always disable the service in rgmanager first. Start a service: **rg_test test /etc/cluster/cluster.conf start service *servicename*** Stop a service: **rg_test test /etc/cluster/cluster.conf stop service *servicename***
Calculate and display the resource tree delta between two cluster.conf files.	**rg_test delta *cluster.conf file 1* *cluster.conf file 2*** For example: **rg_test delta /etc/cluster/cluster.conf.bak /etc/cluster/cluster.conf**

Cluster Service Resource Check and Failover Timeout

This appendix describes how **rgmanager** monitors the status of cluster resources, and how to modify the status check interval. The appendix also describes the **__enforce_timeouts** service parameter, which indicates that a timeout for an operation should cause a service to fail.

Note

To fully comprehend the information in this appendix, you may require detailed understanding of resource agents and the cluster configuration file, **/etc/cluster/cluster.conf**. For a comprehensive list and description of **cluster.conf** elements and attributes, refer to the cluster schema at **/usr/share/system-config-cluster/misc/cluster.ng**, and the annotated schema at **/usr/share/doc/system-config-cluster-X.Y.ZZ/cluster_conf.html** (for example **/usr/share/doc/system-config-cluster-1.0.57/cluster_conf.html**).

E.1. Modifying the Resource Status Check Interval

rgmanager checks the status of individual resources, not whole services. (This is a change from **clumanager** on Red Hat Enterprise Linux 3, which periodically checked the status of the whole service.) Every 10 seconds, rgmanager scans the resource tree, looking for resources that have passed their "status check" interval.

Each resource agent specifies the amount of time between periodic status checks. Each resource utilizes these timeout values unless explicitly overridden in the **cluster.conf** file using the special **<action>** tag:

```
<action name="status" depth="*" interval="10" />
```

This tag is a special child of the resource itself in the **cluster.conf** file. For example, if you had a file system resource for which you wanted to override the status check interval you could specify the file system resource in the **cluster.conf** file as follows:

```
<fs name="test" device="/dev/sdb3">
  <action name="status" depth="*" interval="10" />
  <nfsexport...>
  </nfsexport>
</fs>
```

Some agents provide multiple "depths" of checking. For example, a normal file system status check (depth 0) checks whether the file system is mounted in the correct place. A more intensive check is depth 10, which checks whether you can read a file from the file system. A status check of depth 20 checks whether you can write to the file system. In the example given here, the **depth** is set to *****, which indicates that these values should be used for all depths. The result is that the **test** file system is checked at the highest-defined depth provided by the resource-agent (in this case, 20) every 10 seconds.

E.2. Enforcing Resource Timeouts

There is no timeout for starting, stopping, or failing over resources. Some resources take an

indeterminately long amount of time to start or stop. Unfortunately, a failure to stop (including a timeout) renders the service inoperable (failed state). You can, if desired, turn on timeout enforcement on each resource in a service individually by adding **__enforce_timeouts="1"** to the reference in the **cluster.conf** file.

The following example shows a cluster service that has been configured with the **__enforce_timeouts** attribute set for the **netfs** resource. With this attribute set, then if it takes more than 30 seconds to unmount the NFS file system during a recovery process the operation will time out, causing the service to enter the failed state.

```
</screen>
<rm>
  <failoverdomains/>
  <resources>
    <netfs export="/nfstest" force_unmount="1" fstype="nfs" host="10.65.48.65"
           mountpoint="/data/nfstest" name="nfstest_data" options="rw,sync,soft"/>
  </resources>
  <service autostart="1" exclusive="0" name="nfs_client_test" recovery="relocate">
    <netfs ref="nfstest_data" __enforce_timeouts="1"/>
  </service>
</rm>
```

E.3. Changing Consensus Timeout

The consensus timeout specifies the time (in milliseconds) to wait for consensus to be achieved before starting a new round of membership configuration.

When consensus is calculated automatically, the following rules will be used:

- If configuring a cluster of 2 or less nodes, consensus will be **(token * 0.2)** , with a maximum of 2000 milliseconds and a minimum of 200 milliseconds.
- If configuring a cluster of 3 or more nodes, consensus will be **(token + 2000 milliseconds)**

If you let **cman** configure your consensus timeout in this fashion, realize that moving from 2 to 3 (or more) nodes will require a cluster restart, since the consensus timeout will need to change to the larger value based on the token timeout.

When configuring a 2-member cluster with the ultimate intention of adding more nodes at a later time, you must adjust the consensus timeout so that you do not have to restart the cluster to add the new nodes. To do this, you can edit the **cluster.conf** as follows:

```
<totem token="X" consensus="X + 2000" />
```

Note that the configuration parser does not calculate **X + 2000** automatically. An integer value must be used rather than an equation.

The advantage of the optimized consensus timeout for 2 node clusters is that overall failover time is reduced for the 2 node case since consensus is not a function of the token timeout.

Note

For two node auto-detection in **cman**, the number of physical nodes matters and not the presence of the **two_node=1** directive in **cluster.conf**.

High Availabilty LVM (HA-LVM)

The Red Hat High Availability Add-On provides support for high availability LVM volumes (HA-LVM) in a failover configuration. This is distinct from active/active configurations enabled by the Clustered Logical Volume Manager (CLVM), which is a set of clustering extensions to LVM that allow a cluster of computers to manage shared storage.

When to use CLVM or HA-LVM should be based on the needs of the applications or services being deployed.

- If the applications are cluster-aware and have been tuned to run simultaneously on multiple machines at a time, then CLVM should be used. Specifically, if more than one node of your cluster will require access to your storage which is then shared among the active nodes, then you must use CLVM. CLVM allows a user to configure logical volumes on shared storage by locking access to physical storage while a logical volume is being configured, and uses clustered locking services to manage the shared storage. For information on CLVM, and on LVM configuration in general, refer to *Logical Volume Manager Administration*.
- If the applications run optimally in active/passive (failover) configurations where only a single node that accesses the storage is active at any one time, you should use High Availability Logical Volume Management agents (HA-LVM).

Most applications will run better in an active/passive configuration, as they are not designed or optimized to run concurrently with other instances. Choosing to run an application that is not cluster-aware on clustered logical volumes may result in degraded performance if the logical volume is mirrored. This is because there is cluster communication overhead for the logical volumes themselves in these instances. A cluster-aware application must be able to achieve performance gains above the performance losses introduced by cluster file systems and cluster-aware logical volumes. This is achievable for some applications and workloads more easily than others. Determining what the requirements of the cluster are and whether the extra effort toward optimizing for an active/active cluster will pay dividends is the way to choose between the two LVM variants. Most users will achieve the best HA results from using HA-LVM.

HA-LVM and CLVM are similar in the fact that they prevent corruption of LVM metadata and its logical volumes, which could otherwise occur if multiple machines where allowed to make overlapping changes. HA-LVM imposes the restriction that a logical volume can only be activated exclusively; that is, active on only one machine at a time. This means that only local (non-clustered) implementations of the storage drivers are used. Avoiding the cluster coordination overhead in this way increases performance. CLVM does not impose these restrictions - a user is free to activate a logical volume on all machines in a cluster; this forces the use of cluster-aware storage drivers, which allow for cluster-aware file systems and applications to be put on top.

HA-LVM can be setup to use one of two methods for achieving its mandate of exclusive logical volume activation.

- The preferred method uses CLVM, but it will only ever activate the logical volumes exclusively. This has the advantage of easier setup and better prevention of administrative mistakes (like removing a logical volume that is in use). In order to use CLVM, the High Availability Add-On and Resilient Storage Add-On software, including the **`clvmd`** daemon, must be running.

  The procedure for configuring HA-LVM using this method is described in Section F.1, “Configuring HA-LVM Failover with CLVM (preferred, Red Hat Enterprise Linux 5.6 and later)”.
- The second method uses local machine locking and LVM "tags". This method has the advantage of not requiring any LVM cluster packages; however, there are more steps involved in setting it up and it does not prevent an administrator from mistakenly removing a logical volume from a node in the cluster where it is not active. The procedure for configuring HA-LVM using this method is

described in Section F.2, “Configuring HA-LVM Failover with Tagging”.

F.1. Configuring HA-LVM Failover with CLVM (preferred, Red Hat Enterprise Linux 5.6 and later)

To set up HA-LVM failover (using the preferred CLVM variant), perform the following steps:

1. Ensure that your system is configured to support CLVM, which requires the following:
   - The High Availability Add-On and Resilient Storage Add-On are installed, including the the **cmirror** package if the CLVM logical volumes are to be mirrored.
   - The **locking_type** parameter in the global section of the **/etc/lvm/lvm.conf** file is set to the value '3'.
   - The High Availability Add-On and Resilient Storage Add-On software, including the **clvmd** daemon, must be running. For CLVM mirroring, the **cmirrord** service must be started as well.
2. Create the logical volume and file system using standard LVM and file system commands, as in the following example.

```
# pvcreate /dev/sd[cde]1

# vgcreate -cy shared_vg /dev/sd[cde]1

# lvcreate -L 10G -n ha_lv shared_vg

# mkfs.ext3 /dev/shared_vg/ha_lv

# lvchange -an shared_vg/ha_lv
```

   For information on creating LVM logical volumes, refer to *Logical Volume Manager Administration*.

3. Edit the **/etc/cluster/cluster.conf** file to include the newly created logical volume as a resource in one of your services. Alternately, you can use **Conga** or the **ccs** command to configure LVM and file system resources for the cluster. The following is a sample resource manager section from the **/etc/cluster/cluster.conf** file that configures a CLVM logical volume as a cluster resource:

```
<rm>
   <failoverdomains>
       <failoverdomain name="FD" ordered="1" restricted="0">
          <failoverdomainnode name="neo-01" priority="1"/>
          <failoverdomainnode name="neo-02" priority="2"/>
       </failoverdomain>
   </failoverdomains>
   <resources>
       <lvm name="lvm" vg_name="shared_vg" lv_name="ha-lv"/>
       <fs name="FS" device="/dev/shared_vg/ha-lv" force_fsck="0"
force_unmount="1" fsid="64050" fstype="ext3" mountpoint="/mnt" options=""
self_fence="0"/>
   </resources>
   <service autostart="1" domain="FD" name="serv" recovery="relocate">
```

```
        <lvm ref="lvm"/>
        <fs ref="FS"/>
    </service>
</rm>
```

F.2. Configuring HA-LVM Failover with Tagging

To set up HA-LVM failover by using tags in the **/etc/lvm/lvm.conf** file, perform the following steps:

1. Ensure that the **locking_type** parameter in the global section of the **/etc/lvm/lvm.conf** file is set to the value '1'.

2. Create the logical volume and file system using standard LVM and file system commands, as in the following example.

```
# pvcreate /dev/sd[cde]1

# vgcreate shared_vg /dev/sd[cde]1

# lvcreate -L 10G -n ha_lv shared_vg

# mkfs.ext3 /dev/shared_vg/ha_lv
```

   For information on creating LVM logical volumes, refer to *Logical Volume Manager Administration*.

3. Edit the **/etc/cluster/cluster.conf** file to include the newly created logical volume as a resource in one of your services. Alternately, you can use **Conga** or the **ccs** command to configure LVM and file system resources for the cluster. The following is a sample resource manager section from the **/etc/cluster/cluster.conf** file that configures a CLVM logical volume as a cluster resource:

```
<rm>
    <failoverdomains>
        <failoverdomain name="FD" ordered="1" restricted="0">
            <failoverdomainnode name="neo-01" priority="1"/>
            <failoverdomainnode name="neo-02" priority="2"/>
        </failoverdomain>
    </failoverdomains>
    <resources>
        <lvm name="lvm" vg_name="shared_vg" lv_name="ha_lv"/>
        <fs name="FS" device="/dev/shared_vg/ha_lv" force_fsck="0"
force_unmount="1" fsid="64050" fstype="ext3" mountpoint="/mnt" options=""
self_fence="0"/>
    </resources>
    <service autostart="1" domain="FD" name="serv" recovery="relocate">
        <lvm ref="lvm"/>
        <fs ref="FS"/>
    </service>
</rm>
```

Note

If there are multiple logical volumes in the volume group, then the logical volume name (**lv_name**) in the **lvm** resource should be left blank or unspecified. Also note that in an HA-LVM configuration, a volume group may be used by only a single service.

4. Edit the **volume_list** field in the **/etc/lvm/lvm.conf** file. Include the name of your root volume group and your hostname as listed in the **/etc/cluster/cluster.conf** file preceded by @. The hostname to include here is the machine on which you are editing the **lvm.conf** file, not any remote hostname. Note that this string *MUST* match the node name given in the **cluster.conf** file. Below is a sample entry from the **/etc/lvm/lvm.conf** file:

```
volume_list = [ "VolGroup00", "@neo-01" ]
```

   This tag will be used to activate shared VGs or LVs. *DO NOT* include the names of any volume groups that are to be shared using HA-LVM.

5. Update the **initrd** device on all your cluster nodes:

```
# mkinitrd -f /boot/initrd-$(uname -r).img $(uname -r)
```

6. Reboot all nodes to ensure the correct **initrd** device is in use.

Upgrading A Red Hat Cluster from RHEL 4 to RHEL 5

This appendix provides a procedure for upgrading a Red Hat cluster from RHEL 4 to RHEL 5. The procedure includes changes required for Red Hat GFS and CLVM, also. For more information about Red Hat GFS, refer to *Global File System: Configuration and Administration*. For more information about LVM for clusters, refer to *LVM Administrator's Guide: Configuration and Administration*.

Upgrading a Red Hat Cluster from RHEL 4 to RHEL 5 consists of stopping the cluster, converting the configuration from a GULM cluster to a CMAN cluster (only for clusters configured with the GULM cluster manager/lock manager), adding node IDs, and updating RHEL and cluster software. To upgrade a Red Hat Cluster from RHEL 4 to RHEL 5, follow these steps:

1. Stop client access to cluster high-availability services.
2. At each cluster node, stop the cluster software as follows:
    a. Stop all high-availability services.
    b. Run **service rgmanager stop**.
    c. Run **service gfs stop**, if you are using Red Hat GFS.
    d. Run **service clvmd stop**, if CLVM has been used to create clustered volumes.

Note

If **clvmd** is already stopped, an error message is displayed:

```
# service clvmd stop
Stopping clvm:
[FAILED]
```

The error message is the expected result when running **service clvmd stop** after **clvmd** has stopped.

    e. Depending on the type of cluster manager (either CMAN or GULM), run the following command or commands:
        - CMAN — Run **service fenced stop; service cman stop**.
        - GULM — Run **service lock_gulmd stop**.
    f. Run **service ccsd stop**.
3. Disable cluster software from starting during reboot. At each node, run **/sbin/chkconfig** as follows:

```
# chkconfig --level 2345 rgmanager off
# chkconfig --level 2345 gfs off
# chkconfig --level 2345 clvmd off
# chkconfig --level 2345 fenced off
# chkconfig --level 2345 cman off
# chkconfig --level 2345 ccsd off
```

4. Edit the cluster configuration file as follows:

a. At a cluster node, open **/etc/cluster/cluster.conf** with a text editor.

b. If your cluster is configured with GULM as the cluster manager, remove the GULM XML elements — **<gulm>** and **</gulm>** — and their content from **/etc/cluster/cluster.conf**. GULM is not supported in Red Hat Cluster Suite for RHEL 5. Example G.1, "GULM XML Elements and Content" shows an example of GULM XML elements and content.

c. At the **<clusternode>** element for each node in the configuration file, insert **nodeid="*number*"** after **name="*name*"**. Use a *number* value unique to that node. Inserting it there follows the format convention of the **<clusternode>** element in a RHEL 5 cluster configuration file.

Note

The **nodeid** parameter is required in Red Hat Cluster Suite for RHEL 5. The parameter is optional in Red Hat Cluster Suite for RHEL 4. If your configuration file already contains **nodeid** parameters, skip this step.

d. When you have completed editing **/etc/cluster/cluster.conf**, save the file and copy it to the other nodes in the cluster (for example, using the **scp** command).

5. If your cluster is a GULM cluster and uses Red Hat GFS, change the superblock of each GFS file system to use the DLM locking protocol. Use the **gfs_tool** command with the **sb** and **proto** options, specifying **lock_dlm** for the DLM locking protocol:

   gfs_tool sb *device* proto lock_dlm

   For example:

```
# gfs_tool sb /dev/my_vg/gfs1 proto lock_dlm
You shouldn't change any of these values if the filesystem is mounted.

Are you sure? [y/n] y

current lock protocol name = "lock_gulm"
new lock protocol name = "lock_dlm"
Done
```

6. Update the software in the cluster nodes to RHEL 5 and Red Hat Cluster Suite for RHEL 5. You can acquire and update software through Red Hat Network channels for RHEL 5 and Red Hat Cluster Suite for RHEL 5.

7. Run **lvmconf --enable-cluster**.

8. Enable cluster software to start upon reboot. At each node run **/sbin/chkconfig** as follows:

```
# chkconfig --level 2345 rgmanager on
# chkconfig --level 2345 gfs on
# chkconfig --level 2345 clvmd on
# chkconfig --level 2345 cman on
```

9. Reboot the nodes. The RHEL 5 cluster software should start while the nodes reboot. Upon verification that the Red Hat cluster is running, the upgrade is complete.

Example G.1. GULM XML Elements and Content

```
<gulm>
  <lockserver name="gulmserver1"/>
  <lockserver name="gulmserver2"/>
  <lockserver name="gulmserver3"/>
</gulm>
```

Revision History

Revision 10.0-8 **Mon Sep 8 2014** **Steven Levine**
Version for 5.11 GA release

Revision 10.0-5 **Mon Jun 30 2014** **Steven Levine**
Beta release of Red Hat Enterprise Linux 5.11

Revision 10.0-3 **Tue Jun 10 2014** **Steven Levine**
Resolves: #571695
Notes that Red Hat Enterprise Linux 5 supports bonding mode 1 only.

Resolves: #480291
Improves document index

Resolves: #1092636
Adds description of Startup Wait parameter to description of PostgreSQL 8 fields.

Revision 10.0-2 **Wed Apr 30 2014** **Steven Levine**
Latest draft for RHEL 5.11 release

Revision 9.0-6 **Mon Sep 30 2013** **Steven Levine**
Version for Red Hat Enterprise Linux 5.10 GA release

Revision 9.0-5 **Wed Jul 10 2013** **Steven Levine**
Beta release of Red Hat Enterprise Linux 5.10

Revision 9.0-4 **Tue May 28 2013** **Steven Levine**
Resolves: #960841
Documents need to set SELinux booleans for fence_xvm fencing agent operation.

Revision 9.0-3 **Tue May 21 2013** **Steven Levine**
Incorporating review comment re: note about SELinux.

Revision 9.0-1 **Mon May 20 2013** **Steven Levine**
Clarifies usage of SELinux with virtual machines.

Revision 8.0-6 **Fri Jan 4 2013** **Steven Levine**
Version for Red Hat Enterprise Linux 5.9 GA release

Revision 8.0-5 **Wed Sep 26 2012** **Steven Levine**
Fixing typographical error

Revision 8.0-3 **Wed Aug 29 2012** **Steven Levine**
Beta release of Red Hat Enterprise Linux 5.9

Resolves: #823649
Documents new attribute for file system resource agents to set NFS workarounds.

Revision 8.0-1 **Tue Jul 31 2012** **Steven Levine**

Resolves: #742312
Documents support for IBM iPDU Fence Device

Resolves: #757902
Documents backup and restore of luci configuration using **luci_admin** command.

Resolves: #838154
Documents virtual machine resource configuration.

Resolves: #831343
Updates descriptions of luci fields and screens.

Resolves: #838131
Documents SSL attribute for HP iLO fence device.

Resolves: #712379
Adds an appendix on configuring high availability LVM.

Revision 7.0-3 Thu Feb 16 2012 Steven Levine

Release for GA of Red Hat Enterprise Linux 5.8

Resolves: #712376
Adds information on disabling cluster software.

Resolves: #712387
Adds information on stopping single resources of a cluster.

Resolves: #712593
Adds appendix on consensus timeout.

Resolves: #626495
Adds note on single site cluster support.

Revision 7.0-2 Thu Dec 15 2011 Steven Levine

Beta release of Red Hat Enterprise Linux 5.8

Revision 7.0-1 Thu Nov 10 2011 Steven Levine

Resolves: #571557
Adds note on managing virtual machines in a cluster.

Resolves: #742310
Documents new privilege level parameter for IPMI fence device.

Resolves: #747456
Corrects small typographical errors throughout document.

Resolves: #748935
Clarifies description of iptables firewall filters.

Resolves: #718084
Provides link to Support Essentials article.

Resolves: #749858
Documents support for RHEV-M REST API fence agent.

Resolves: #569585
Clarifies support statement for running Samba in a cluster.

Revision 6.0-1	**Thu Jul 21 2011**	**Steven Levine**

Resolves: #713256
Documents new fence_vmware_soap agent.

Resolves: #446137
Documents procedure to configure a system to listen to luci from the internal network only.

Resolves: #515858
Provides information about cluster service status check and failover timeout.

Resolves: #560558
Provides rules to allow multicast traffic for cluster comunication

Resolves: #705131
Updates tables of fence agent parameters to reflect Red Hat Enterprise Linux 5.7 support.

Resolves: #705134
Documents non-critical resources and restart-disable configuration parameter.

Resolves: #480292
Adds pointer to cluster.conf schema in documentation of resource parameters.

Resolves: #515860
Updates example domains.

Resolves: #595711
Fixes minor typographical errors.

Resolves: #654176
Fixes minor typographical errors.

Resolves: #675809
Fixes incorrect table title reference.

Revision 5.0-1 **Thu Dec 23 2010** **Steven Levine**

Resolves: #664055
Adds newly-supported fence agents to Fence Device Parameters appendix.

Revision 4.0-1 **Mon Mar 15 2010** **Paul Kennedy**

Resolves: #511150
Clarifies support for SELinux.

Resolves: #527473
Adds information about cluster node-count limit.

Resolves: #568179
Adds information about support of and GFS/GFS2 deployment.

Resolves: #568483
Adds general support statement.

Resolves: #526540
Clarifies meaning of Name parameter for fencing devices.

Revision 3.0-1	Tue Aug 18 2009	Paul Kennedy

Resolves: #516128
Adds notes about not supporting IPV6.

Resolves: #482936
Corrects Section 5.7 title and intro text.

Resolves: #488751
Corrects iptables rules. Removed examples.

Resolves: #502053
Corrects iptables rules for rgmanager.

Resolves: #511150
Adds content stating that SELinux must be disabled for Red Hat Cluster Suite.

Resolves: #513072
Adds information about limitations on using SCSI reservations as a fencing method.

Revision 2.0-1	Tue Jan 20 2009	Paul Kennedy

Resolves: #458882
Explains Firewall settings for multicast address.

Resolves: #450777
Includes content about configuring failover domains to not fail back a service (an added feature).

Revision 1.0-1	Wed May 21 2008	Michael Hideo Smith

Resolves: #232215
Changing from XML to HTML Single with floating Table of Contents and viewable by browser

Index

A

ACPI
- configuring, Configuring ACPI For Use with Integrated Fence Devices

Apache HTTP Server
- httpd.conf , Installing and Configuring the Apache HTTP Server
- setting up service, Example of Setting Up Apache HTTP Server

Apache server resource agent, HA Resource Parameters

APC power switch, Fence Device Parameters

APC power switch over SNMP, Fence Device Parameters

B

behavior, HA resources, HA Resource Behavior

Brocade fabric switch, Fence Device Parameters

Bull PAP (Platform Administration Processor), Fence Device Parameters

C

Cisco MDS, Fence Device Parameters

Cisco UCS, Fence Device Parameters

cluster
- administration, Before Configuring a Red Hat Cluster, Managing Red Hat Cluster With Conga, Managing Red Hat Cluster With system-config-cluster
- diagnosing and correcting problems, Diagnosing and Correcting Problems in a Cluster, Diagnosing and Correcting Problems in a Cluster
- disabling the cluster software, Disabling the Cluster Software
- displaying status, Cluster Status Tool, Managing High-Availability Services
- managing node, Managing Cluster Nodes
- starting, Starting the Cluster Software
- starting, stopping, restarting, and deleting, Starting, Stopping, and Deleting Clusters

cluster administration, Before Configuring a Red Hat Cluster, Managing Red Hat Cluster With Conga, Managing Red Hat Cluster With system-config-cluster
- backing up the cluster database, Backing Up and Restoring the Cluster Database
- compatible hardware, Compatible Hardware
- configuring ACPI, Configuring ACPI For Use with Integrated Fence Devices
- configuring iptables, Enabling IP Ports
- configuring max_luns, Configuring max_luns
- Conga considerations, Considerations for Using Conga
- considerations for using qdisk, Considerations for Using Quorum Disk
- considerations for using quorum disk, Considerations for Using Quorum Disk
- diagnosing and correcting problems in a cluster, Diagnosing and Correcting Problems in a Cluster, Diagnosing and Correcting Problems in a Cluster
- disabling the cluster software, Disabling the Cluster Software
- displaying cluster and service status, Cluster Status Tool, Managing High-Availability Services
- enabling IP ports, Enabling IP Ports
- general considerations, General Configuration Considerations
- managing cluster node, Managing Cluster Nodes
- managing high-availability services, Managing High-Availability Services
- modifying the cluster configuration, Modifying the Cluster Configuration
- network switches and multicast addresses, Multicast Addresses
- restoring the cluster database, Backing Up and Restoring the Cluster Database
- SELinux, Red Hat Cluster Suite and SELinux
- starting and stopping the cluster software, Starting and Stopping the Cluster Software
- starting, stopping, restarting, and deleting a cluster, Starting, Stopping, and Deleting Clusters
- virtual machines, Configuring Virtual Machines in a Clustered Environment

cluster configuration, Configuring Red Hat Cluster With Conga
- modifying, Modifying the Cluster Configuration

Cluster Configuration Tool
- accessing, Cluster Configuration Tool

cluster database
- backing up, Backing Up and Restoring the Cluster Database
- restoring, Backing Up and Restoring the Cluster Database

cluster resource relationships, Parent, Child, and Sibling Relationships Among Resources

cluster resource status check, Cluster Service Resource Check and Failover Timeout

cluster resource types, Considerations for Configuring HA Services

cluster service
- displaying status, Cluster Status Tool, Managing High-Availability Services

cluster service managers
- configuration, Adding a Cluster Service to the Cluster, Adding a Cluster Service to the Cluster, Propagating The Configuration File: New Cluster

cluster services, Adding a Cluster Service to the Cluster, Adding a Cluster Service to the Cluster
- (see also adding to the cluster configuration)
- Apache HTTP Server, setting up, Example of Setting Up Apache HTTP Server
  - httpd.conf , Installing and Configuring the Apache HTTP Server

cluster software
- configuration, Configuring Red Hat Cluster With Conga
- disabling, Disabling the Cluster Software
- installation and configuration, Configuring Red Hat Cluster With system-config-cluster
- starting and stopping, Starting and Stopping the Cluster Software

cluster software installation and configuration, Configuring Red Hat Cluster With system-config-cluster

cluster storage
- configuration, Configuring Cluster Storage

command line tools table, Command Line Administration Tools

configuration
- HA service, Considerations for Configuring HA Services

configuration file
- propagation of, Propagating The Configuration File: New Cluster

configuring cluster storage , Configuring Cluster Storage

Configuring High Availability LVM, High Availabilty LVM (HA-LVM)

Conga
- accessing, Configuring Red Hat Cluster Software
- considerations for cluster administration, Considerations for Using Conga
- overview, Conga

Conga overview, Conga

D

Dell DRAC, Fence Device Parameters

E

Egenera SAN controller, Fence Device Parameters

F

failover timeout, Cluster Service Resource Check and Failover Timeout

feedback, Feedback

fence device
- APC power switch, Fence Device Parameters
- APC power switch over SNMP, Fence Device Parameters
- Brocade fabric switch, Fence Device Parameters
- Bull PAP (Platform Administration Processor), Fence Device Parameters
- Cisco MDS, Fence Device Parameters
- Cisco UCS, Fence Device Parameters
- Dell DRAC, Fence Device Parameters
- Egenera SAN controller, Fence Device Parameters
- Fujitsu Siemens Remoteview Service Board (RSB), Fence Device Parameters
- GNBD (Global Network Block Device), Fence Device Parameters
- HP iLO (Integrated Lights Out), Fence Device Parameters
- HP iLO (Integrated Lights Out) MP, Fence Device Parameters
- IBM Blade Center, Fence Device Parameters
- IBM iPDU, Fence Device Parameters
- IBM Remote Supervisor Adapter II (RSA II), Fence Device Parameters
- IF MIB, Fence Device Parameters
- IPMI (Intelligent Platform Management Interface) LAN, Fence Device Parameters
- manual fencing, Fence Device Parameters
- McData SAN switch, Fence Device Parameters
- QLogic SANBox2 switch, Fence Device Parameters
- RHEV-M REST API, Fence Device Parameters
- RPS-10 power switch, Fence Device Parameters
- SCSI fencing, Fence Device Parameters
- virtual machine fencing, Fence Device Parameters
- Vixel SAN switch, Fence Device Parameters
- VMware (SOAP interface), Fence Device Parameters
- WTI power switch, Fence Device Parameters

filesystem resource agent, HA Resource Parameters

Fujitsu Siemens Remoteview Service Board (RSB), Fence Device Parameters

G

general
- considerations for cluster administration, General Configuration Considerations

GFS file system resource agent, HA Resource Parameters

GNBD (Global Network Block Device), Fence Device Parameters

H

HA service configuration
- overview, Considerations for Configuring HA Services

hardware
- compatible, Compatible Hardware

HP iLO (Integrated Lights Out), Fence Device Parameters

HP iLO (Integrated Lights Out) MP, Fence Device Parameters

HTTP services
- Apache HTTP Server
  - httpd.conf, Installing and Configuring the Apache HTTP Server
  - setting up, Example of Setting Up Apache HTTP Server

I

IBM Blade Center, Fence Device Parameters

IBM iPDU, Fence Device Parameters

IBM Remote Supervisor Adapter II (RSA II), Fence Device Parameters

IF MIB, Fence Device Parameters

integrated fence devices
- configuring ACPI, Configuring ACPI For Use with Integrated Fence Devices

introduction, Introduction
- other Red Hat Enterprise Linux documents, Introduction

IP address resource agent, HA Resource Parameters

IP ports
- enabling, Enabling IP Ports

IPMI (Intelligent Platform Management Interface) LAN, Fence Device Parameters

iptables
- configuring, Enabling IP Ports

iptables firewall, Configuring the iptables Firewall to Allow Cluster Components

L

LVM resource agent, HA Resource Parameters

LVM, High Availability, High Availabilty LVM (HA-LVM)

M

manual fencing, Fence Device Parameters

max_luns
- configuring, Configuring max_luns

McData SAN switch, Fence Device Parameters

multicast addresses
- considerations for using with network switches and multicast addresses, Multicast Addresses

multicast traffic, enabling, Configuring the iptables Firewall to Allow Cluster Components

MySQL resource agent, HA Resource Parameters

N

NFS client resource agent, HA Resource Parameters

NFS export resource agent, HA Resource Parameters

NFS mount resource agent, HA Resource Parameters

O

open LDAP resource agent, HA Resource Parameters

Oracle 10g failover instance resource agent, HA Resource Parameters

Oracle DB resource agent, HA Resource Parameters

Oracle listener resource agent, HA Resource Parameters

P

parameters, fence device, Fence Device Parameters

parameters, HA resources, HA Resource Parameters

PostgresSQL 8 resource agent, HA Resource Parameters

Q

qdisk
- considerations for using, Considerations for Using Quorum Disk

QLogic SANBox2 switch, Fence Device Parameters

quorum disk
- considerations for using, Considerations for Using Quorum Disk

R

relationships
- cluster resource, Parent, Child, and Sibling Relationships Among Resources

resource agent
- Apache server, HA Resource Parameters
- filesystem, HA Resource Parameters
- GFS file system, HA Resource Parameters
- IP address, HA Resource Parameters
- LVM, HA Resource Parameters
- MySQL, HA Resource Parameters
- NFS client, HA Resource Parameters
- NFS export, HA Resource Parameters
- NFS mount, HA Resource Parameters
- open LDAP, HA Resource Parameters
- Oracle 10g failover instance, HA Resource Parameters
- Oracle DB, HA Resource Parameters
- Oracle listener, HA Resource Parameters
- PostgresSQL 8, HA Resource Parameters
- Samba service, HA Resource Parameters
- SAP database, HA Resource Parameters
- SAP instance, HA Resource Parameters
- Sybase ASE failover instance, HA Resource Parameters
- Tomcat 5, HA Resource Parameters

RHEV-M REST API, Fence Device Parameters

RPS-10 power switch, Fence Device Parameters

S

Samba service resource agent, HA Resource Parameters

SAP database resource agent, HA Resource Parameters

SAP instance resource agent, HA Resource Parameters

SCSI fencing, Fence Device Parameters

SELinux
- configuring, Red Hat Cluster Suite and SELinux

starting the cluster software, Starting the Cluster Software

status check, cluster resource, Cluster Service Resource Check and Failover Timeout

Sybase ASE failover instance resource agent, HA Resource Parameters

System V init , Starting and Stopping the Cluster Software

T

table
- command line tools, Command Line Administration Tools

tables
- HA resources, parameters, HA Resource Parameters

timeout failover, Cluster Service Resource Check and Failover Timeout

Tomcat 5 resource agent, HA Resource Parameters

troubleshooting
- diagnosing and correcting problems in a cluster, Diagnosing and Correcting Problems in a Cluster, Diagnosing and Correcting Problems in a Cluster

types
- cluster resource, Considerations for Configuring HA Services

U

upgrading, RHEL 4 to RHEL 5, Upgrading A Red Hat Cluster from RHEL 4 to RHEL 5

V

virtual machine fencing, Fence Device Parameters

virtual machine resource service, HA Resource Parameters

virtual machines, in a cluster, Configuring Virtual Machines in a Clustered Environment

Vixel SAN switch, Fence Device Parameters

VMware (SOAP interface), Fence Device Parameters

W

WTI power switch, Fence Device Parameters